CMPE 257 – Homework1

Exercise 1.3:

The weight update rule in (1.3) has the nice interpretation that it moves in the direction of classifying correctly.

(a) Show that $y(t)w^T(t)x(t) < 0$. [Hint: x(t) is misclassified by w(t).]

Solution:

Since, x(t) is misclassified by w(t) so:

1. for y(t) = +1, sign($w^T(t)x(t)$) = -1

So, $w^T(t)x(t) < 0$

Therefore, $y(t)\, w^T(t)x(t) < 0$

2. for y(t) = -1, sign($w^T(t)x(t)$) = +1

So, $w^T(t)x(t) > 0$

Therefore, $y(t)\, w^T(t)x(t) < 0$

(b) Show that $y(t)w(t+1)\, x(t) > y(t)\, w^T(t)x(t)$. [Hint: Use (1.3).]

Solution:

From previous solution, we know that Since, x(t) is misclassified by w(t).

$y(t)\, w^T(t)x(t) < 0$

We know that for a misclassified point, when weight is adjusted, the new iteration of weight vector will classify the point better than the previous iteration as it always moves in the right direction.

Therefore, $y(t)w(t+1)\, x(t) > y(t)\, wT(t)x(t)$

(c) As far as classifying x(t) is concerned, argue that the move from w(t) to w(t+1) is a move “in the right direction”.

Solution:

Initially with random weight we check the sign of the inner product of weight vector and input vector depending on formula A.B = ||A|| ||B|| cosϴ

If the point is misclassified, the weight vector is updated.

$$\mathbf{W_1 = W} + y_N \mathbf{X_N}$$

Here the weight is adjusted depending on the misclassified point both in terms of X_N and y_N { -1, +1}. This new weight will cause the output to be more likely correctly classified. As the number of iterations increase, the point is more and more likely to be correctly classified.

Exercise 1.6:

For each of the following tasks, identify which type of learning is involved (supervised, reinforcement or unsupervised) and the training data to be used. If a task can fit more than one type, explain how and describe the training data for each type.

(a) Recommending a book to a user in an online bookstore – We use **supervised learning** to recommend a book to a user as it is based off the books that he has purchased earlier. The training data used in this scenario will be the historical data of the user.

(b) Playing tic tac toe – The game tic tac toe uses **reinforcement learning** to learn the game using the outcome of previous game and by making mistakes, until the algorithm figures out what step to take in which scenario to have a successful outcome. The training data used here is the games played before the present one

(c) Categorizing movies into different types – This can be performed using supervised and unsupervised learning. In case of **supervised learning**, the training data set has previous data which helps in classifying the movies

based on genre and depending on comedy content, action and other factors. Another way of learning can be through **unsupervised learning** wherein the training data is there but its unlabeled. Through unsupervised learning, patterns and structures can be found in the data to categorize into different types.

(d) Learning to play music – This is another example of **reinforcement learning**, where a person plays music, learns from the outcome and improves his technique of playing music based on that. The training data is the music played before and its mistakes to improve learning

(e) Credit limit: Deciding the maximum allowed debt for each bank customer – This is a problem which can be solved using linear regression model of the **supervised learning** algorithm. The training data is the historical data that the bank has of different customers having various parameters. Based on the past data, an algorithm is created to calculate the credit limit for a customer.

Exercise 1.8: If $\mu = 0.9$, what is the probability that a sample of 10 marbles will have $\nu <= 0.1$? [Hints: 1. Use binomial distribution. 2. The answer is very small number.]

Solution:

μ = probability of red marbles in bin

ν = fraction of red marbles in sample.

$P(\nu <= 0.1) = P(red/N <= 0.1)$

$\Leftrightarrow P(red <= 0.1N) = P(red <=1)$

$P(red <=1) = P(red =0) + P(red =1)$

P(red) = 0.9

P(green) = 1 -0.9 = 0.1

Using binomial distribution,

$P(x=k) = \binom{n}{k} p^k (1-p)^{n-k}$

$P(red = 0) = \binom{10}{0} 0.9^0 0.1^{10} = (10!/0! * 10!) * 0.1^{10} = 10^{-10}$

$P(red = 1) = \binom{10}{1} 0.9^{10} 0.1^9 = (10!/1!*9!) * 0.9^{10} 0.1^9 = 9*10^{-9}$

$P(\nu <= 0.1) = 10^{-10} + 9*10^{-9} = 9.1 *10^{-9}$

Exercise 1.9: If μ = 0.9, use the hoeffding inequality to bound the probability that a sample of 10 marbles will have ν <= 0.1 and compare the answer to the previous exercise.

Solution:

μ = probability of red marbles in bin

ν = fraction of red marbles in sample.

ν <= 0.1 => ν - μ <= 0.1- 0.9

- ⇨ ν - μ <= -0.8
- ⇨ | ν - μ | >= 0.8^{-}, which means that |ν - μ| is slightly less than 0.8

Therefore Epsilon, ϵ= 0.8^{-}

Using Hoeffding inequality,

$P[|v - \mu| > \epsilon] <= 2e^{-2\epsilon 2N}$

$P[|v - \mu| > 0.8^{-}] <= 2e^{-2 .8*.8 *10}$

$\approx 5.52 *10^{-6}$

Using The hoeffding inequality, we are capable of bounding the probability of $| v - \mu |$ by $2e^{-2\epsilon 2N}$ which depends only on the size of the bin and not on μ. The bin can be large or small, finite or infinite, and we will still get the same bound when we use the same sample size.

We can verify the bound using the output of exercise 1.8 where
$P(v <= 0.1) <= P(|v - \mu| > 0.8^{-})$
As, $9.1 * 10^{-9} <= 5.52 *10^{-6}$

Problem 1.2: Consider the perceptron in two dimensions: $h(x) = sign(w^T x)$ where $w = [w_0, w_1, w_2]^T$ and $x = [1, x_1, x_2]^T$. Technically, x has three coordinates, but we call this perceptron two-dimensional because the first coordinate is fixed at 1.

(a) Show that the regions on the plane where h(x) = +1 and h(x) = -1 are separated by a line. If we express this line by the equation x2 = ax1+b, what are the slope a and intercept b in terms of w_0, w_1, w_2?

(b) Draw a picture of the cases $w =[1,2,3]^T$ and $w = -[1,2,3]^T$.

In more than 2 dimensions, the +1 and -1 regions are separated by a hyperplane, the generalization of a line.

Solution:

(a) For a two-dimensional perceptron,

$H(x) = sign(w^T x)$

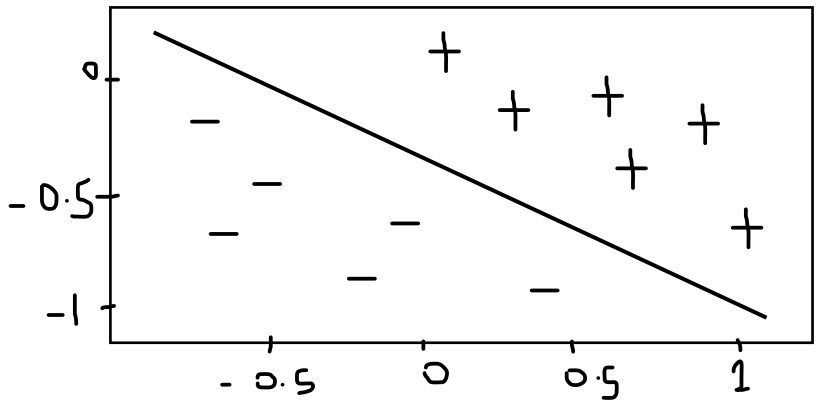
For h(x) = +1, sign($w^T x$)> 0 and similarly for h(x) =-1, sign($w^T x$) <0. Separation between these two regions can be derived from the equation $w^T x = 0$

$w^T x = w_0 + w_1 x_1 + w_2 x_2$

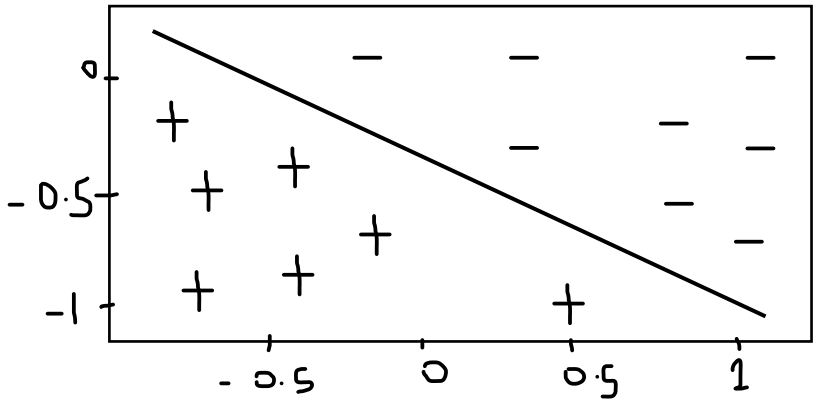
- ⇨ $w_0 + w_1 x_1 + w_2 x_2 = 0$, where w_0 is the bias
- ⇨ $x_2 = -w_0/w_2 - w_1/w_2 * x_1$

From the above equation, we can see that,
$a = - w_0/w_2$
$b = - w_1/w_2$

(b) For $w = [1,2,3]^T$, the equation of line will be
$1 + 2x + 3y = 0$. Plot of the line is shown below:



For $w = [1,2,3]^T$, the equation of line will be
$-1-2x-3y = 0$. The plot of the line will be the same, but the data points will be classified differently:



The regions for h(x) = +1 and h(x) = -1 will be reversed in the second case

---

Problem 1.4: In jupyter notebook

Problem 1.5: in jupyter notebook

Problem 1.11: The matrix which tabulates the cost of various errors for the CIA and supermarket applications in example 1.1 is called a risk or loss matrix. For the two risk matrices in example 1.1, explicitly write down the in-sample error Ein that one should minimize to obtain g. This in-sample error should weight the different types of errors based on the risk matrix. [Hint: Consider yn =+1 and yn =-1 separately.]

Solution:

For the supermarket example, the matrix is shown below:

		f	
		+1	-1
h	+1	0	1
	-1	10	0

$E_{in}(h) = 1/N \sum_{n=1}^{N} e(h(xn), f(xn))$

$= 1/N \left[\sum_{yn=1} e(h(xn), 1) + \sum_{yn=-1}^{N} e(h(xn), -1) \right]$

$= 1/N \left[\sum_{yn=1} 10[h(xn) \neq 1] + \sum_{yn=-1}^{N} [h(xn) \neq -1] \right]$

The pointwise error will be

$e(h(x_n), y_n) = 10$, if $h(x) = -1$ and $y_n = +1$

$= 1$, if $h(x) = 1$ and $y_n = -1$

$= 0$, otherwise

For the CIA example, the matrix is shown below:

		f	
		+1	-1
h	+1	0	1000
	-1	1	0

$E_{in}(h) = 1/N \sum_{n=1}^{N} e(h(xn), f(xn))$

$= 1/N \left[\sum_{yn=1} e(h(xn), 1) + \sum_{yn=-1}^{N} e(h(xn), -1) \right]$

$= 1/N \left[\sum_{yn=1} [h(xn) \neq 1] + \sum_{yn=-1}^{N} 1000.[h(xn) \neq -1] \right]$

The pointwise error will be

$e(h(x_n), y_n) =$ 1000 , if $h(x) = 1$ and $y_n = -1$

= 1 , if $h(x) = -1$ and $y_n = 1$

= 0 , otherwise

The E_{in} needs to be minimized to obtain g.