Baye's rule

Suppose we have one red and one blue box. In the red box we have 2 apples and 6 oranges, whilst in the blue box we have 3 apples and 1 orange. Now suppose we randomly selected one of the boxes and picked a fruit. If the picked fruit is an orange, what is the probability that it was picked from the blue box?

From the above question, we are required to find the probability that given we picked an orange, it came from the blue box. We are provided with the following information,

Probability of picking a fruit from the red Red Box, P(Red)= 0.6
Probability of picking a fruit from the Blue Box P(Blue) = 1-P(Red) = 0.4
Probability of picking an orange given the box is Red - P(Orange|Red) = 3/4 = 0.75
Probability of picking an orange given the box is Blue - P(Orange|Blue) = 1 - 1/4 = 0.25

$$P(Blue|Orange) = \frac{P(Blue) * P(Orange|Blue)}{P(Blue) * P(Orange|Blue) + P(Red) * P(Orange|Red)}$$

$$P(Blue|Orange) = \frac{0.4 * 0.25}{0.4 * 0.25 + 0.6 * 0.75} = 0.1818$$

Thus the probability of getting an Orange provided we took it from a blue box is 0.1818.

Ridge regression

Batch gradient descent

Given the gradient descent algorithms for linear regression (discussed in Chapter 2 of Module 2), derive weight update steps of stochastic gradient descent (BGD) as well as batch gradient

descent (BGD) for linear regression with L2 regularisation norm.

We perform the following steps for batch gradient descent,

- Intialise the W^0 and t=1
- while a stopping condition is not met do:
  - $\eta^{'}$ = η
  - while $\eta^1 > \epsilon$ do
    - W:= W^{t-1} -$\eta^{'}\nabla E(W^{t-1})$ # we get this from the above derivation
    - if $E(W) < E(W^{t-1})$ then **break**
    - $\eta^{'} = \eta'/2$
  - $W^t = W$
  - t=t+1

To get the weight update steps, we derive it as done below

The objective function for linear regression with L2 regularisation norm is given by the formula given below,

$$E(W) = \frac{1}{2}\sum_{n=1}^{N}[t_n - W.\phi(x_n)]^2 + \frac{\lambda}{2}\sum_{j=0}^{M-1}W^2$$

where $D := (x_n, t_n)_{n=1}^{N}$

We minimize this error function by differentiating E(W) with respect to W. On performing this, we get our gradient as shown below,

$$\nabla E(W) = -\sum_{n=1}^{N}[t_n - W.\phi(x_n)]\phi(x_n) + \lambda W$$

We update the weights as done below,

$$W^t := W^{t-1} - \eta^{'}\nabla E(W^{t-1})$$

This process repeats after processing one batch at a time.

Stochastic gradient descent

We perform the following steps for stochastic gradient descent.

- Intialise the W^0 and t=1
- while a stopping condition is not met do:
  - $\eta^{'} = \eta$
  - while $\eta^1 > \epsilon$ do
    - for each training point (x_n, y_n)
      - $W^t := W^{t-1} - \eta^{'} \nabla E_n(W^{t-1})$
      - t=t+1
    - if $E(W) < E(W^{t-1})$ then **break**
    - $\eta^{'} = \eta^{'}/2$

To get the weight update steps, we derive it as done below,

Similar to BGD, we begin with L2 regularisation norm and differentiate it partially with respect to w to get the gradient. This gradient is then multiplied with eta and subracted from the old weight to update the new weight.

$$E_n(W) = \frac{1}{2}[t_n - W.\phi(x_n)]^2 + \sum_{j=0}^{M-1} \frac{\lambda}{2} W_j^2$$

where $D := (x_n, t_n)_{n=1}^N$ is the training data

The gradient objective obtained by differentiating above equation with respect to W

$$\nabla E_n(W) = -[t_n - W.\phi(x_n)]\phi(x_n) + \sum_{j=0}^{M-1} \lambda W_j$$

W_j corresponds to each data point as stochastic gradient descent performs over each datapoint in the dataset.

We then update our weights as done below,

$$W^t := W^{t-1} - \eta' \nabla E_n(W^{t-1})$$

For each datapoint.

In [ ]: