Probabilistic Machine Learning

Recall the simple example from Appendix A. of Module 1.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? Note that the chance of picking the red box is 40% and the selection chance for any of the pieces from a box is equal for all the pieces in that box.

1. Firstly, we need to denote the all the variable given in the following question like the following:

- Random variable of box = B
- Random variable for fruit = F
- red box and blue box = r & b
- apple and orange = a & o

2. Then, we will denote and calculate the probability value based on the given information as the following:

- Probability of picking red box: $p(B{=}r)$ = 40% or 0.4 or $\frac{40}{100}$

regarding the provided information we can calculate the probability og picking blue box by 1 - $p(B{=}r)$ we will get $p(B{=}b)$ = 0.6

Next, we will find the probability of getting each fruit by the following:

- sample from red box have 2 apples and 6 oranges:2+6= 8
- sample from blue box have 3 apples and 1 oranges:3+1= 4

From above, we will get $p(a|r)$ = $\frac{2}{8}$ or 0.25, $p(o|r)$ = $\frac{6}{8}$ or 0.75, $p(a|b)$ = $\frac{3}{4}$ or 0.75, and $p(o|b)$ = $\frac{1}{4}$ or 0.25, which is the probability of getting apple given/depend on red box, the probability of getting orange given/depend on red box and vice versa.

3. We will used all of the following value to calculate probability of picking blue box given an orange using a bayes theorem to calculate the probability as below:

Identify what is $p(b|o)$ by the following procedure:

$p(b|o)$ = $\frac{p(o|b)*p(b)}{p(o)}$

*Note $p(b|o)$ is a likelihood of picking blue box given an orange

Since, we already identify $p(b)$ and $p(o|b)$ we need to identify the $p(o)$ by using the following method:

When you got an unknown probability, in this case, probability of orange will be express as the following: $p(o)$ = $p(o|b)p(b){+}p(o|not\ b)p(not\ b)$

from the following, we will get this bayes theorem formula, then we substitute the value and get the answer as the following:

$p(b|o){=}\frac{p(o|b)*p(b)}{p(o|b)*p(b)+p(o|not\ b)*p(not\ b)}$ = $\frac{0.25*0.6}{0.25*0.6+0.75*0.4}$ = $\frac{0.15}{0.45}$ = 0.333

Therefore, we will get the probability of picking up the blue box given orange is 0.333

Reference

The approach for getting an answer is derieved from:

- Chen, B. (2022). *Week 2.:Probabilistic Machine Learning* [PowerPoint slides]. https://lms.monash.edu/mod/resource/view.php?id=9894962 (https://lms.monash.edu/mod/resource/view.php?id=9894962)
- Haffari, G. (2018, July 3rd). *The Elements of Machine Learning*. https://lms.monash.edu/mod/resource/view.php?id=10054436 (https://lms.monash.edu/mod/resource/view.php?id=10054436)

5 Ridge Regression Question 5. part I.

SGD & Batch Gradient Descent purpose is to optimize objective function (loss/error function)

SGD for ridge regression:

1. Initialising of $w^{(0)}$ or weight vector of parameters, picking eta(η) or learning rate as termination criteria, and epsilon (threshold). $\{(0,0),(5,0)\ldots(x_n,y_n)\}$

2. Identify the loss function of the ridge regression, which is the same as linear regression adding L2 regularize term as the following:

$$E = \frac{1}{2}\sum_{n=1}^{N}(y - w \cdot \phi(x_n))^2 + \frac{\lambda}{2}\sum_{j=0}^{m-1} w_j^2$$

$$= \frac{1}{2}\sum_{n=1}^{N}(y - w\,\phi(x_n))^2 + \frac{\lambda}{2} w^T \cdot w$$

Then, we will identify the gradient of this function for using in the SGD loop when updating the weight by derivation of loss function as the following:

$$E\nabla = \frac{dL}{dw} = -\left(y - (w \cdot \phi(x_n)\right)x_n + \lambda w$$

Once the gradient is identify, it will be used in the iteration for weight updating with the following formula: $w^{(T)} = w^{(T-1)} - \eta E \nabla$

3. When we get all the necessary function, this will show how an SGD work after initiated the vector parameter as the follow

starting point
$t = 1$

While (loop which will keep iterate until it met threshold)

1st iter

It will randomly select the data point eg. (5, 0) from all data point.

Update the vector until the threshold met or termination criteria is complete:

update weight vector formula
$w^{(1)} = w^{(0)} - \eta \cdot -(y - (w^{(0)} \cdot \phi(x_n))) x_n + \lambda w^{(0)}$

Suppose our threshold = 0.1, $\lambda = 0$, $\phi = 2$

$w^{(0)} = 5$

$w^{(1)} = 5 - 0.5 \cdot -(0 - (5 \cdot 2(5)))5 + (0 \cdot 5)$

$= 126$

Based on the update weight, the loop will continue until it found optimal data point

$t = t + 1$ # increase iterate number

Second loop will do the same process until condition is met

Batch Gradient Descent

For batch gradient descent, these are the following steps of how this algorithm works:

- The first step and second step are the same as SGD
- 3rd step: In this algorithm, the loop will iterate through all the datapoints then providing the minimal point which taking quite longer to identify-ing optimal value comparing with SGD.

The loop:

1st loop   From 1st iteration $(t=1)$ to last datapoint:

└ The data point is not randomly picked because it will iterate through all data point

└ If the stopping point not met criteria (weight $\neq$ epsilon)

└ $t = t+1$

└ $w = w^{(t-1)} - \eta \cdot (-(y - w \cdot \phi(x_n)) x_n + \lambda w$

which take a very long time to complete for Large dataset.

Reference

The algorithm implementation is derieved from the following:

- Haffari, G. (2016, July). *CodeBase_A1_Q5 (1).R*. https://lms.monash.edu/pluginfile.php/14028235/mod_assign/intro/CodeBase_A1_Q5%20%281%29.R (https://lms.monash.edu/pluginfile.php/14028235/mod_assign/intro/CodeBase_A1_Q5%20%281%29.R)
- Haffari, G. (2019, January 9th). *Linear Models for Regression*. https://lms.monash.edu/mod/resource/view.php?id=10099576 (https://lms.monash.edu/mod/resource/view.php?id=10099576)

In [ ]: