MONASH University

FIT5202 DATA ANALYSIS ALGORITHMS

Assignment 1

Author: Suyash Sathe
Student Id: 29279208
Email: ssat0005@student.monash.edu
Master of Data Science, Monash University

Section C. Probabilistic Machine Learning

Question 4 [Bayes Rule, 20 Marks]

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 apple, 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.

Solution

Given: Picked fruit is an apple.

To find: Probability that the apple was picked from the blue box, i.e., $P(Blue \mid Apple)$

According to Bayes' Rule,

$$P(Blue \mid Apple) = \frac{P(Apple \mid Blue)\, P(Blue)}{P(Apple)}$$

Where,

$P(Apple \mid Blue) = \frac{3}{4}$

$P(Blue) = \frac{6}{10}$

$$P(Apple) = P(Apple \mid Blue) * P(Blue) + P(Apple \mid Red) * P(Red)$$

$$= \left(\frac{3}{4} \times \frac{6}{10}\right) + \left(\frac{2}{8} \times \frac{4}{10}\right) = \frac{44}{80} = \frac{11}{20}$$

$$\therefore P(Blue \mid Apple) = \frac{\frac{3}{4} \times \frac{6}{10}}{\frac{11}{20}} = \frac{3}{4} \times \frac{6}{10} \times \frac{20}{11} = \frac{9}{11}$$

Thus, $P(Blue \mid Apple) = \frac{9}{11}$

Question 5 [Maximum Likelihood, 20 Marks]

As opposed to a coin which has two faces, a dice has 6 faces. Suppose we are given a dataset which contains the outcomes of 10 independent tosses of a dice: D:={1,4,5,3,1,2,6,5,6,6}. We are asked to build a model for this dice, i.e. a model which tells what the probability of each face of the dice is if we toss it. Using the maximum likelihood principle, please determine the best value for our model parameters.

Solution

Given: D: $= \{1, 4, 5, 3, 1, 2, 6, 5, 6, 6\}$

Since the dice has 6 faces, let the probability associated with each face (1, 2, 3, 4, 5, 6) be $\omega 1, \omega 2, \omega 3, \omega 4, \omega 5\ and\ \omega 6$.

Probability of ωi where $i \in \{1,6\}$ is $0 \leq \omega i \leq 1$

Applying the maximum likelihood principle,

$p(D \mid \omega 1) = \prod_{i=1}^{10} p(x_i \mid \omega 1) = \omega 1^2 (1-\omega 1)^8$, where x_i represents every event in the dataset.

$$\log(p(D \mid \omega 1)) = \log \omega 1^2 + \log(1-\omega 1)^8$$

$$\frac{d\log(p(D \mid \omega 1))}{d\omega 1} = \frac{2}{\omega 1} - \frac{8}{(1-\omega 1)}$$

$$0 = \frac{2}{\omega 1} - \frac{8}{(1-\omega 1)}$$

$$\frac{2}{\omega 1} = \frac{8}{(1-\omega 1)}$$

$$10\omega 1 = 2$$

$$\therefore\ \omega 1 = 0.2$$

Similarly,

$$p(D \mid \omega 2) = \prod_{i=1}^{10} p(x_i \mid \omega 2) = \omega 2(1-\omega 2)^9$$

$$\log(p(D \mid \omega 2)) = \log \omega 2 + \log(1-\omega 2)^9$$

$$\frac{d\log(p(D \mid \omega 2))}{d\omega 2} = \frac{1}{\omega 2} - \frac{9}{(1-\omega 2)}$$

$$0 = \frac{1}{\omega 2} - \frac{9}{(1-\omega 2)}$$

$$\frac{1}{\omega 2} = \frac{9}{(1-\omega 2)}$$

$$10\omega 2 = 1$$

$$\therefore\ \omega 2 = 0.1$$

$$p(D \mid \omega 3) = \prod_{i=1}^{10} p(x_i \mid \omega 3) = \omega 3(1-\omega 3)^9$$

$$\log(p(D \mid \omega 3)) = \log \omega 3 + \log(1-\omega 3)^9$$

$$\frac{d\log(p(D \mid \omega 3))}{d\omega 3} = \frac{1}{\omega 3} - \frac{9}{(1-\omega 3)}$$

$$0 = \frac{1}{\omega 3} - \frac{9}{(1-\omega 3)}$$

$$\frac{1}{\omega 3} = \frac{9}{(1-\omega 3)}$$

$$10\omega 3 = 1$$

$$\therefore\ \omega 3 = 0.1$$

$$p(D \mid \omega 4) = \prod_{i=1}^{10} p(x_i \mid \omega 4) = \omega 4(1-\omega 4)^9$$

$$\log(p(D \mid \omega 4)) = \log \omega 4 + \log(1-\omega 4)^9$$

$$\frac{d\log(p(D \mid \omega 4))}{d\omega 4} = \frac{1}{\omega 4} - \frac{9}{(1-\omega 4)}$$

$$0 = \frac{1}{\omega 4} - \frac{9}{(1-\omega 4)}$$

$$\frac{1}{\omega 4} = \frac{9}{(1-\omega 4)}$$

$$10\omega 4 = 1$$

$$\therefore\ \omega 4 = 0.1$$

$$p(D \mid \omega 5) = \prod_{i=1}^{10} p(x_i \mid \omega 5) = \omega 5^2(1-\omega 5)^8$$

$$\log(p(D \mid \omega 5)) = \log \omega 5^2 + \log(1-\omega 5)^8$$

$$\frac{d\log(p(D \mid \omega 5))}{d\omega 5} = \frac{2}{\omega 5} - \frac{8}{(1-\omega 5)}$$

$$0 = \frac{2}{\omega 5} - \frac{8}{(1-\omega 5)}$$

$$\frac{2}{\omega 5} = \frac{8}{(1-\omega 5)}$$

$$10\omega 5 = 2$$

$$\therefore \ \omega 5 = 0.2$$

$$p(D \mid \omega 6) = \prod_{i=1}^{10} p(x_i \mid \omega 6) = \omega 6^3(1-\omega 6)^7$$

$$\log(p(D \mid \omega 6)) = \log \omega 6^3 + \log(1-\omega 6)^7$$

$$\frac{d\log(p(D \mid \omega 6))}{d\omega 6} = \frac{3}{\omega 6} - \frac{7}{(1-\omega 6)}$$

$$0 = \frac{3}{\omega 6} - \frac{7}{(1-\omega 6)}$$

$$\frac{3}{\omega 6} = \frac{7}{(1-\omega 6)}$$

$$10\omega 6 = 3$$

$$\therefore \ \omega 6 = 0.3$$

Thus, the best values of our model parameters are $\omega 1 = 0.2,\ \omega 2 = 0.1,\ \omega 3 = 0.1,\ \omega 4 = 0.1,\ \omega 5 = 0.2\ and\ \omega 6 = 0.3.$

Section D. Ridge Regression

Question 6 [Ridge Regression, 25 Marks]

Given the gradient descent algorithms for linear regression (discussed in Chapter 2 of Module 2), derive weight update steps of stochastic gradient descent (SGD) as well as batch gradient descent (BGD) for linear regression with L2 regularisation norm. Show your work with enough explanation in your PDF report; you should provide the steps of SGD and BGD, separately.

Solution

Derivation of weight update steps for SGD and BGD

For Ridge Regression,

$$Objective\ function\ =\ Error\ +\ Regularization$$

$$Objective\ function\ = E(w) + \lambda.\ \Omega(w)$$

Where,

$E(w) \rightarrow error\ term$

$\lambda \rightarrow regularization\ parameter$

$\Omega(w) \rightarrow penalty\ \ (regularization\ function)$

For Linear Regression,

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - w.\emptyset(x_n))^2$$

For L2 Regularization, regularization function,

$$\Omega(w) = \frac{\lambda}{2}\sum_{j=0}^{M-1} w^2$$

Therefore,

$$Objective\ function = \frac{1}{2}\sum_{n=1}^{N}(t_n - w.\emptyset(x_n))^2 + \frac{\lambda}{2}\sum_{j=0}^{M-1} w^2$$

Now, gradient of the objective function,

$$\nabla(Objective\ function) = \nabla\left(\frac{1}{2}\sum_{n=1}^{N}\left(t_n - w.\emptyset(x_n)\right)^2 + \frac{\lambda}{2}\sum_{j=0}^{M-1} w^2\right)$$

$$= \nabla\left(\frac{1}{2}\sum_{n=1}^{N}\left(t_n - w.\emptyset(x_n)\right)^2\right) + \nabla\left(\frac{\lambda}{2}\sum_{j=0}^{M-1} {w_j}^2\right)$$

$$\therefore \nabla(Objective\ function) = \left[-\sum_{n=1}^{N}\left(t_n - w.\emptyset(x_n)\right)\right].\emptyset(x_n) + \lambda.w$$

where, $\sum_{j=0}^{M-1} w_j = w$

For SGD and BGD,

$$new\ parameter = old\ parameter - \eta\nabla(Objective\ function)$$

where, $\eta\ is\ the\ learning\ rate$.

Therefore,

$$w^{(T)} = w^{(T-1)} - \eta^{(T)}\nabla(Objective\ function)$$

$$\therefore w^{(T)} = w^{(T-1)} - \eta^{(T)}\left(\left[-\sum_{n=1}^{N}\left(t_n - w^{(T-1)}.\emptyset(x_n)\right)\right].\emptyset(x_n) + \lambda.w^{(T-1)}\right)$$

Therefore, $w^{(T)}$ is the updated weight for stochastic gradient descent (SGD) as well as batch gradient descent (BGD) for linear regression with L2 regularisation norm.

Stochastic Gradient Descent (SGD)

In SGD algorithm, the weight parameters are calculated and updated for each row/value (X_i) of the training data. The weight parameter is updated as per the above derivation of $w^{(T)}$. This updated weight parameter is used to update the next weight parameter. This process is continued till the termination condition is reached. Following is the algorithm for the above steps:

1. Initialize the weight parameters to some random initial values.
$$\mathrm{w}^{(1)} = [\mathrm{w}_0, \mathrm{w}_1, \mathrm{w}_2, \dots, \mathrm{w}_\mathrm{m}]$$
2. Plug the parameter values into the weight update function for every point in the dataset to get the gradient ($\nabla(Objective\ function)$).
3. Calculate the step size: $\eta.\nabla(Objective\ function)$

$$step\ size = \eta^{(1)} \left(\left[-\sum_{n=1}^{N} \left(t_n - w^{(0)}.\emptyset(x_n) \right) \right].\emptyset(x_n) + \lambda.w^{(0)} \right)$$

4. Calculate the new parameter:

$$new\ parameter = old\ parameter - step\ size$$

$$new\ parameter = old\ parameter - \eta.\nabla(Objective\ function)$$

5. Repeat steps 3 to 5 until the step size becomes very small ,i.e., the algorithm converges, or maximum number of iterations are performed.

In this algorithm, the dataset is resampled to get a batch of the dataset and for every data point in the batch, the weight parameters are updated until it converges, or threshold of iterations is reached.

Batch Gradient Descent (BGD)

In the BGD algorithm, the weight parameters are calculated and updated using entire dataset as a batch and not every data point in the dataset. The calculation of weight parameters is similar to SGD, but instead of every point in the dataset, the gradient is calculated using the mean of all data points (X_i) in the batch. Following is the algorithm for the above steps:

1. Initialize the weight parameters to some random initial values like SGD.

$$\mathrm{w}^{(1)} = [\mathrm{w}_0, \mathrm{w}_1, \mathrm{w}_2, \dots, \mathrm{w}_m]$$

2. Plug the parameter values into the weight update function and use the mean of predicted values and true values from the dataset to get the gradient ($\nabla(Objective\ function)$).

   (Note that mean of true and predicted values for the dataset is taken and not the individual data points).

3. Calculate the step size: $\eta.\nabla(Objective\ function)$

$$step\ size = \eta \left(\left[-\sum_{n=1}^{N} \left(mean(t_n) - w^{(0)}.mean(\emptyset(x_n)) \right) \right].mean(\emptyset(x_n)) + \lambda.w^{(0)} \right)$$

4. Calculate the new parameter:

$$new\ parameter = old\ parameter - step\ size$$

5. Repeat steps 3 to 5 until the step size becomes very small ,i.e., the algorithm converges, or maximum number of iterations are performed.

Thus, the implementation and function of SGD and BGD is quite similar. The only difference is that SGD calculates weight parameters for each point in the dataset whereas BGD considers the mean of the batch of the dataset.