Assignment-1 Report

Question 1:

III

i.) Optimum value for K in terms of the testing error:

We notice that testing and training errors converge between K values 8 & 9. Therefore, we consider K = 9 to be the most optimum value for the testing error.

ii.) Discuss the values of K and model complexity corresponding to underfitting and overfitting:

We notice that for K = 1, The model's training error is at its lowest (i.e, 0) while the test error is very high. Therefore, we can assume that the model is too complicated and is hence overfitting.
The testing error keeps increasing gradually until k = 2 and thereafter starts dropping. The difference between the training and testing error is high until K = 5 and reduces sharply after. Hence, we can assume the model is overfitting until K = 5.

Beyond K = 9, both the testing and training errors are very high and increase sharply. This indicates that for very high values of K, the model is overly simplistic and exhibits underfitting.

Question 2:

III

The optimum value for K:

We observe that the average error is least when K = 4 and gradually rises after. Therefore, we consider it to be the most optimum value.

Compare your result with III of Q1 and comment on the ability of the CV to select a good model:

Since L-fold Cross validation allows us to employ multiple and different datasets for training and testing, it generates a better generalisation when compared with a simple KNN regressor. From the plot we observe that the L-fold Cross validation technique gives us the most optimal model when K = 4 when compared to the KNN regressor (K = 9). This is due to better generalisation of the errors.

Question 3:

III

Based on the plot in the previous part (Part Π), how does the test error and its uncertainty behave as K increases?

We notice that for lower values of K, the test error and its uncertainty are small as the variance is small and the distribution of data exhibits little to no skew. However, as the value of K increases, the variance increases correspondingly. For large values of K, we have very high variance and we notice that the data is positively skewed from K = 4. Therefore, test error is significantly more for high values of K when compared with low values.

V

Based on the plot in the previous part (Part IV), how does the test error and its uncertainty behave as the number of subsets in bootstrapping increases?

We notice that for lower number of subsets, the mean error is negatively skewed and the model has high variance. However, from Times = 60, the mean error is positively skewed. Further, the mean error is high for lower number of subsets, whereas, it reduces significantly for when we have higher number of subsets and the variance is very low. This is appropriate as having more number of subsets allow us to improve the model.

Question 4:

Given,

Red box has 3 apples and 5 oranges,

The blue box we have 4 apple and 4 oranges, and

The yellow box we have 1 apple and 1 orange.

Probability of picking red box,

Pr(Red Box) = $\frac{1}{2}$

Pr(Blue Box) = $\frac{3}{10}$

Pr(Yellow Box) = $\frac{1}{5}$

To find,

Pr(Yellow Box| apple)

By Bayes Theorem we have,

$$\Pr(Yellow\ Box|\ apple)$$

$$= \frac{\Pr(apple|\ Yellow\ Box) * \Pr(Yellow\ Box)}{\Pr(apple|\ Yellow\ Box) * \Pr(Yellow\ Box) + \Pr(apple|Yellow\ Box') * \Pr(Yellow\ Box')}$$

Since, Yellow box has 1 apple and 1 orange,

Pr(apple|Yellow Box) = $\frac{1}{2}$

$\Pr(apple|Yellow\ Box') * \Pr(Yellow\ Box')$ = Pr(apple|Blue Box)*Pr(Blue Box) + Pr(apple|Red Box)*Pr(Red Box)

Where, Pr(apple|Blue Box) = $\frac{4}{8} = \frac{1}{2}$ (Since the blue box has 4 apples and 4 oranges)

Pr(apple|Red Box) = $\frac{3}{5}$ (Since the red box has 3 apples and 5 oranges)

Therefore,

$\Pr(apple|Yellow\ Box') * \Pr(Yellow\ Box')$ = $\frac{1}{2} * \frac{3}{10} + \frac{3}{5} * \frac{1}{2}$ = 0.45

$\Pr(Yellow\ Box|\ apple)$ = $\frac{\frac{1}{2}*\frac{1}{5}}{0.45}$ = 0.2222

Question 5:

Given the gradient descent algorithms for linear regression (discussed in Chapter 2 of Module 2), derive weight update steps of stochastic gradient descent (SGD) for linear regression with L2 regularisation norm.

Answer:

The weight vector for the next iteration tau +1 is given by,

$w^{\tau+1} = w^{\tau} - \eta^{\tau+1}(\nabla E).\, w^{\tau}$

Where $\nabla E = (t - w^{\tau}.\phi_n)$

the data dependent error term for regression is usually the sum of squares error function i.e,

$$E(D) = \frac{1}{2}\sum_{n=1}^{\mathbb{N}} (t_n - w \cdot \phi(x_n))^2$$

which is also rooted in maximum likelihood. Putting the error function and the regularisation term together, the resulting training objective is:

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

Ridge regression corresponds to the case where q =2.

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

In order to minimise the training objective, we set its partial derivative to 0.

$$\frac{\mathrm{dE(w)}}{\mathrm{dw}} = \frac{d}{dw}[\frac{1}{2}\sum_{n=1}^{N}(t_n - w \cdot \phi(x_n))^2 + \frac{\lambda}{2}\sum_{j=0}^{M-1} w_j^2]$$

$$\frac{\mathrm{dE(w)}}{\mathrm{dw}} = \sum_{n=1}^{N}(t_n - w \cdot \phi(x_n)\,(-\phi(x_n) + \lambda\sum_{j=0}^{M-1} w_j^1 = 0$$

We update the weight as,

$$w^{(\tau+1)} := w^{(\tau)} + \eta^{(\tau+1)}\sum_{n=1}^{N}((t_n - w \cdot \phi(x_n))(\phi(x_n)) - \eta^{(\tau+1)}\lambda\sum_{j=0}^{M-1} w_j$$

Based on your plot in the previous part (Part b), what's the best value for lambda? Discuss lambda, model complexity, and error rates, corresponding to underfitting and overfitting, by observing your plot.

Answer:

Based on the plot, we can determine the best lambda by determining the point on the plot where the testing error is at the least possible value. We observe that the least testing error value is at log(lambda) = -1.2 i.e we consider lambda value of 0.4.

We can say that a model is underfitting when the testing and training errors are high. From the plot we observe that, the testing and training errors are low

for low values of lambda and high for high values of lambda. But, both the testing and training errors are high for approx. log(lambda) = 1.

We can say that a model is overfitting when the testing error is high and training error is low. From the plot we observe that, the training error is low and testing error is high when log(lambda) = i.e, lambda = . Therefore,

The error rates are for the plot have the following trend: For low values of lambda i.e lambda values in the range: , the testing and training are low but steadily rising.

Question 6:

For each class, train an individual perceptron ("one-vs-all", i.e., using one designated class as positive and all others as negative) and compute the test error resulting from combining the weight vectors of those k perceptrons to form an alternative multiclass classifier (that predicts using the argmax rule as in the code above). Comment on whether this approach is better or worse than the one implemented in III.

Answer:

We notice from the plots that in the case of multi class perceptron, the test error is initially high but drops for the later sets of mini batches whereas in the case of one vs All perceptron, the test error fluctuates a lot and remains higher when compared to multi class perceptron. Therefore, this approach is worse off than Multi class perceptron.

Conceptually, one vs All classification is not viable when large number of classes are present and the data is huge. Therefore, Multi Class perceptron is better as it can handle large number of classes for large data sets.

Question 7:

a. **What happens for each classifier when the number of training data points is increased?**

Answer:

When the number of data points are increased, we notice that the training and testing errors reduce and converge.

For logistic regression classifier, the training error is initially similar when compared with Bayesian classifier, but as the number of data points increase, we notice that the training error for both mostly converge and for some cases such as sample_size = 260, 270, 290 , 340 and 480, we notice that the Bayesian classifier returns lower training errors.

When we compare, testing errors, both the classifiers initially show an increase in error rate and then drop gradually from sample size =60 and converge for the most part. But, when sample size is very high, we notice that for a brief period, the logistic regression classifier, returns slightly larger testing error.

b. Which classifier is best suited when the training set is small, and which is best suited when the training set is big?

Answer: For small datasets, both the classifiers perform equally good but for bigger training sets, both converge for most parts but the Bayesian classifier performs better as logistic regression classifier occasionally returns higher errors.

c. Justify your observations in previous questions (III.a & III.b) by providing some speculations and possible reasons.

Answer: The training and testing errors for both classifiers are mostly similar because the data set provided has only 2 classes. Hence, it is very difficult to gain accurate predictions with each classifier and analyse their performance.