Assignment 1

Deadline: 5th February, 11:55 PM

1 Introduction

1.1 Bias-Variance trade off

Whenever we discuss model prediction, it's important to understand prediction errors (bias and variance). There is a trade-off between a model's ability to minimize bias and variance. Gaining a proper understanding of these errors would help to distinguish a layman and an expert in Machine Learning. So, instead of playing around with a number of classifiers, let's understand how to select which classifier to use.

Let's get started and understand some of the basic definitions. For basic definitions when $\hat{f}$ is applied to an unseen sample x refer here.

- **Bias** is the difference between the average prediction of our model and the correct value which we are trying to predict. A model with high bias does not generalize the data well and oversimplifies the model. It always leads to a high error on training and test data.

$$Bias^2 = (E[\hat{f}(x)] - f(x))^2$$

  where $f(x)$ represents the true value, $\hat{f}(x)$ represents the predicted value

- **Variance** is the variability of a model prediction for a given data point. Again, imagine you can repeat the entire model building process multiple times. The variance is how much the predictions for a given point vary between different realizations of the model.

$$Variance = E\left[(\hat{f}(x) - E[\hat{f}(x)])^2\right]$$

  where $f(x)$ represents the true value, $\hat{f}(x)$ represents the predicted value

- **Noise** is a unwanted distortion in data. Noise is anything that is spurious and extraneous to the original data, that is not intended to be present in the first place, but was introduced due to faulty capturing process.

- **Irreducible error** is the error that can't be reduced by creating good models. It is a measure of the amount of noise in our data. Here it is important to understand that no matter how good we make our model, our data will have certain amount of noise or irreducible error that can not be removed.

$$E[(f(x) - \hat{f}(x))^2] = Bias^2 + \sigma^2 + Variance$$
$$\sigma^2 = E[(f(x) - \hat{f}(x))^2] - (Bias^2 + Variance)$$

  where $f(x)$ represents the true value, $\hat{f}(x)$ represents the predicted value, $E[(f(x)-\hat{f}(x))^2]$ is the mean squared error and σ^2 represents irreducible error.

If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand, if our model has a large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

1.2 Liner Regression

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog). There are two main types:

- Simple regression
- Multivariable regression

For a more detailed definition refer this article

For a simple linear regression model with only one feature the equation is:

$$y = w_1 x + b$$

where,

- y =Predicted value/Target Value
- x =Input
- w_1 =Gradient/slope/Weight
- b =Bias

For a Multivariable regression model the equation is:

$$y = b + \sum_{i=1}^{n} w_1 x_1$$

Once we have the prediction function we need to determine the value of weight/s and bias. To see how to calculate the value of weight/s and bias refer this article

2 Tasks

2.1 Task 1: Linear Regression

Write a brief about what function does the method, *LinearRegression().fit()* performs.

2.2 Task 2: Calculating Bias and Variance

In this question you are going to calculate the bias and variance of your trained model.

2.2.1 How to Re-sample data

You have been provided with two datasets i.e., train set and test set, consisting of pairs (xi; yi). It can be loaded into your python program using pickle.load() function. Now divide the train set into 10 equal parts randomly, so that you will get 10 different train datasets to train your model.

2.2.2 Task

After re-sampling data, you have 11 different datasets (10 train sets and 1 test set). Train a linear classifier on each of the 10 train set separately, so that you have 10 different classifiers or models. Now you can calculate the bias and variance of the model using the test set. You need to repeat the above process for the following class of functions,

- $y = mx + c$
- $y = ax^2 + bx + c$
- $y = ax^3 + bx^2 + cx + d$

And so on up till polynomial of degree 20. **You are only supposed to use sklearn's linear_model.LinearRegression().fit() and preprocessing.PolynomialFeatures() for finding the appropriate coefficients with the default parameters.** Tabulate the values of bias and variance and also write a detailed report explaining how bias and variance changes as you vary your function classes.

Note: Whenever we are talking about the bias and variance of model, it refers to the average bias and variance of the model over all the test points.

2.3 Task 3: Calculating Irreducible Error

Tabulate the values of irreducible error for the models in Task 2 and also write a detailed report explaining why or why not the value of irreducible error changes as you vary your class function.

2.4 Task 4: Plotting $Bias^2 - Variance$ graph

Task: Based on the variance, bias and total error calculated in earlier tasks, plot the $Bias^2 - Variance$ tradeoff graph and write your observations in the report with respect to underfitting, overfitting and also comment on the type of data just by analyzing the $Bias^2 - Variance$ plot.

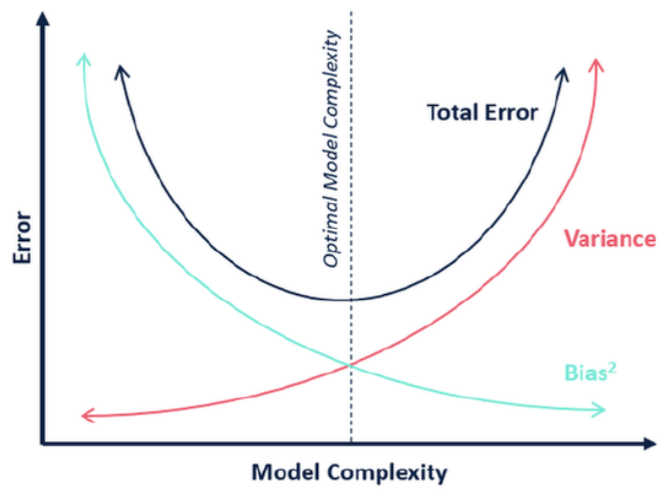

Figure 1: A balance between model framework error and model complexity

Note: The formula for $Bias^2$ and $Variance$ are for a single input but as the testing data contains more than one input, take the mean wherever required.

3 General Instructions

- The data is in numpy array format
- Assignment is do be done in teams of two. Only one of the member is required to make the submission.
- Submit a zip file name **TeamNumber_assgn1.zip** containing source code and the report.

```
TeamNumber_assgn1
├─ code.ipynb
├─ report.pdf
└─ readme.txt (optional for any assumptions)
```

- All coding has to be done in Python3 only, using **Jupyter Notebook**.
- Report should include all details needed for evaluation. Please include relevant graphs, tables, analysis, observations and writeup as required for each of the tasks above.
- Get familiar with numpy, matplotlib, pickle, pandas dataframe and sklearn.
- You should to write vectorized codes which performs much better compared to individual iteration.
- Plagiarism will be penalized heavily.
- Manual evaluations will be held regarding which further details will be announced later.

4 Marking Scheme

- Task 1: 10%
- Task 2: 30%
- Task 3: 10%
- Task 4: 20%
- Viva: 30%