Machine Data and Learning
Assignment-2
Bias-Variance Trade-off

Maximum Marks: 100
Deadline: 18th February 2025, 11:59pm

1 Introduction

1.1 Bias-Variance trade-off

When we discuss model prediction, it is important to understand the various prediction errors - bias and variance. There is a trade-off between a model's ability to minimise bias and variance. A proper understanding of these errors would help in distinguishing a layman and an expert in Machine Learning. Before using different classifiers, it is important to understand how to select a classifier to use.

Let us get started and understand some basic definitions that are relevant. For basic definitions, when $\hat{f}$ is applied to an unseen sample, refer here.

- **Bias** is the difference between the average prediction of our model and the correct value that we are trying to predict. A model with high bias does not generalise the data well and oversimplifies the model. It always leads to a high error on training and test data. The formula for Bias for a given data point x is:

$$\text{Bias} = E_i[\hat{f}_i(x)] - f(x)$$

  where $f(x)$ represents the true value, $\hat{f}_i(x)$ represents the value predicted by the i-th model, and $E_i[\cdot]$ is the expectation over all models.

- **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 realisations of the model. The formula for Variance for a given data point x is:

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

where $\hat{f}_i(x)$ represents the value predicted by the i-th model, $E_i[\cdot]$ is the expectation over all models.

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

$$E_i\left[(y - \hat{f}_i(x))^2\right] = \text{Bias}^2 + \text{Variance} + \sigma^2$$

$$\implies \sigma^2 = E_i\left[(y - \hat{f}_i(x))^2\right] - (\text{Bias}^2 + \text{Variance})$$

where y represents the true value with some noise, $\hat{f}(x)$ represents the predicted value, $E_i[(\hat{f}(x) - f(x))^2]$ represents the Mean Squared Error (MSE) at a given point x, and σ^2 represents irreducible error.

Note: While reporting the values, you need to take the expectation of Bias, Variance, and MSE over all data points. However, these expected values may not hold the above relation.

but,

$$E_x[\text{MSE}] \neq E_x[\text{Bias}]^2 + E_x[\text{Variance}] + E_x[\sigma^2]$$

$$E_x[\text{MSE}] = E_x[\text{Bias}^2] + E_x[\text{Variance}] + E_x[\sigma^2]$$

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 is going to have high variance and low bias. So we need to find the right (or good) balance without overfitting and underfitting the data.

1.2 Linear 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

- Multi-variable regression

For a more detailed definition, refer to this article.
For simple linear regression with only one feature, the equation is:

$$y = wx + b$$

where:

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

Similarly, for a Multi-variable regression model, the equation is:

$$y = b + \sum_{i=1}^{n} w_i x_i$$

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 to this article.

2 Tasks

Task 1: Gradient Descent

(a) Explain how gradient descent works to find the coefficients when there is one independent variable and one dependent variable.

(b) Extend your explanation to the case of multiple independent variables and one dependent variable. Describe how gradient descent is adapted to handle this scenario and find the coefficients in a multivariate regression model.

Task 2: Numerical on Bias and Variance

In this task, you will be calculating the Bias^2, variance, and MSE for the given dataset and verifying the formula:

$$\text{MSE} = \text{Bias}^2 + \text{Variance}$$

(Assume irreducible error σ^2 to be 0). Consider the dataset (x, y) given by:

$$x : [-3, -1, 0, 2, 3, 4]$$

$$y : [10, 2, 3, 8, 18, 30]$$

On training over different splits of data, you get the following three models:

$$f_1(x) = x^2 + x + 1$$

$$f_2(x) = 2x^2 + 2x + 2$$

$$f_3(x) = x^2 + 2x + 2$$

Calculate the Bias^2, variance, and MSE and verify the above formula:

$$\text{MSE} = \text{Bias}^2 + \text{Variance}$$

Provide detailed steps for the calculations and mention the formulas used.

Task 3: Calculating Bias and Variance

Joshita is managing an online bookstore and wants to analyze book sales. She collected data (x_i, y_i), where x_i represents the number of online ads placed for a book, and y_i represents the number of copies sold. Joshita aims to use this data to predict the sales performance of new books based on ad spending.

By understanding bias and variance, she can choose a model that balances underfitting and overfitting, ensuring accurate sales predictions.

In this task, you will help Joshita evaluate the bias and variance of a trained model, assisting her in optimizing ad strategies for maximum sales.

3.1 How to Re-Sample Data

You are given two datasets, i.e., a training set and a test set, consisting of pairs (x_i, y_i), where x_i corresponds to the profit, and y_i corresponds to the sales. This data can be loaded into your Python program using the `pickle.load()` function.

You then need to divide the training set into 15 equal parts randomly so that you get 15 different training datasets to train your model.

3.2 Task

After re-sampling the data, you have 16 different datasets (15 train sets and 1 test set). Train a linear classifier on each of the 15 train sets separately so that you have 15 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:

- $y = ax + b$

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

And so on, up to polynomials of degree **10**.
The only two functions that you are allowed to use from `sklearn` are:

- `linear_model.LinearRegression().fit()`
- `preprocessing.PolynomialFeatures()`

These functions will help you find the appropriate coefficients with the default parameters.

Tabulate the values of bias and variance and also write a **detailed analysis and observation** explaining how bias and variance change as you vary your function classes from **degree 1 to 10**. Discuss underfitting and overfitting behavior observed across different degrees and explain how model complexity affects the bias-variance tradeoff.

Note: Whenever we are talking about the bias and variance of the model, it refers to the average bias and variance of the model over all the test points. For every degree, Bias and Variance are defined as the mean of biases and variances of all 15 models.

Task 4: Calculating Irreducible Error

Tabulate the values of irreducible error for the models in Task 3 and also write a detailed report explaining why or why not the value of irreducible error changes as you vary your class function.

Task 5: Plotting Bias^2 - Variance Graph

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.

The below figure shows the balance between model framework error and model complexity.

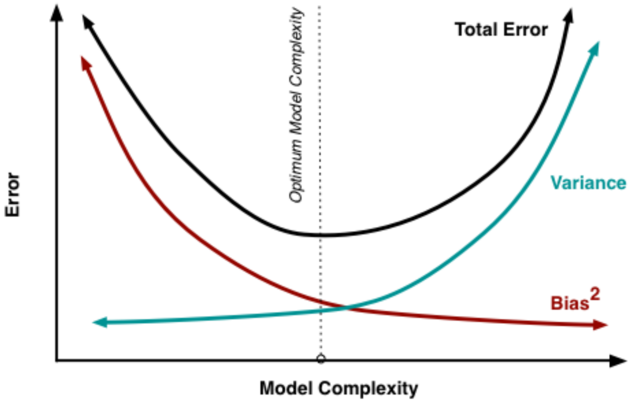

Figure 1: Plot variation of Bias2, Variance and MSE against degree of polynomial in the same graph.

Task 6: Overfitting and Regularization

In this task, you will explore the effects of overfitting and the role of regularization techniques in controlling model complexity.

(a) Fit a polynomial regression model of degree 10 on the given training data.Compute the Mean Squared Error (MSE) on the test set and report the result.

(b) Implement a regularized regression model using either Ridge Regression or Lasso Regression. Train the model with a polynomial of degree 10 using regularization.

(c) Compute and report the MSE on the test set after applying regularization.

(d) Compare the MSE values obtained in parts (a) and (c). Discuss the impact of regularization on overfitting and model performance.

Constraints: You are only allowed to use the following functions from `sklearn`:

- `linear_model.Ridge()`
- `linear_model.Lasso()`

3 General Instructions

- The data is given as a pickle file. On loading the pickle file, you will get a dictionary object that has the training set with the ”train” key and the test set with the ”test” key.
- Submit a zip file named `rollnumber_assgn2.zip` containing the following:
  - `code.ipynb`
  - `report.pdf`
  - `readme.md` (if any assumptions)
- All coding has to be done in Python3 only, using Jupyter Notebook.
- Code is only required for Tasks 3, 4,5 and 6.
- The report should include detailed answers for Tasks 1 and 2.
- The 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`.
- Plagiarism will be penalized heavily.
- Manual evaluations will be held, and further details will be announced later.

4 Marking Scheme

- Task 1: 5 marks
- Task 2: 5 marks
- Task 3: 25 marks
- Task 4: 10 marks
- Task 5: 15 marks
- Task 6: 10 marks
- Viva: 30 marks