Notations

Input Space: $\mathcal{X}$

- The input space contains the set of all possible examples/instances in a **population**.
- This is generally unknown.

Output Space: $\mathcal{Y}$

- The output space is the set of all possible labels/targets that corresponds to each point in $\mathcal{X}$.

Concept: Unknown Ground Truth (Target) Function: $f : \mathcal{X} \to \mathcal{Y}$ where $\mathbf{x} \mapsto \mathbf{y}$ and $\mathbf{y} = f(\mathbf{x})$

- This is a mapping from the input space to the output space. This is the so-called true function which is the underlying true relationship between each pair of point $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$.
- However, this function f is unknown to us, or else we do not need to learn anything. We try our best to find a model that best estimates the **unknown ground truth function** f.

Concept Class: $\mathcal{C}$

- A concept class $\mathcal{C}$ is a set of true functions f. Hypothesis class $\mathcal{H}$ is the set of candidates to formulate as the final output of a learning algorithm to well approximate the true function f. Hypothesis class $\mathcal{H}$ is chosen before seeing the data (training process). $\mathcal{C}$ and $\mathcal{H}$ can be either same or not and we can treat them independently.

Distribution: $\mathcal{P}$

- **Reference: Learning From Data p43.**
- The unknown distribution that generated our input space $\mathcal{X}$.
- In general, instead of the mapping $\mathbf{y} = f(\mathbf{x})$, we can take the output $\mathbf{y}$ to be a random variable that is affected by, rather than determined by, the input $\mathbf{x}$.
- Formally, we have a **target distribution** $P(\mathbf{y}|\mathbf{x})$ instead of just $\mathbf{y} = f(\mathbf{x})$. Now we say that any point $(\mathbf{x}, \mathbf{y})$ in $\mathcal{X}$ is now generated by the **joint distribution**

$$P(\mathbf{x}, \mathbf{y}) = P(\mathbf{x})P(\mathbf{y}|\mathbf{x})$$

Data: $\mathcal{D}$

- This is the set of samples drawn from $\mathcal{X} \times \mathcal{Y}$ over a distribution $\mathcal{P}$.
- The general notation is as follows:

$$\mathcal{D} = [(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})]$$

  where N denotes the number of training samples, and each $\mathbf{x}^{(i)} \in \mathbb{R}^n$ with n features. In general, $\mathbf{y}^{(i)} \in \mathbb{R}$ and is a single label.
- We can split $\mathcal{D}$ into two sets respectively, where $\mathbf{X}$ consists of all the $\mathbf{x}_i$ and $\mathbf{Y}$ consists of all the $\mathbf{y}$. We will see this next.

Design Matrix: $\mathbf{X}$

- Let $\mathbf{X}$ be the design matrix of dimensions $m \times (n+1)$ where m is the number of observations (training samples) and n independent feature/input variables. Note the inconsistency in the matrix size, I just want to point out that the second matrix, has a column of one in the first row because we usually have a bias term $\mathbf{x}_0$, which we set to 1.

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^{(1)})^T \\ (\mathbf{x}^{(2)})^T \\ \vdots \\ (\mathbf{x}^{(m)})^T \end{bmatrix}_{m \times \mathbf{x}} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{bmatrix}_{m \times (n+1)}$$

Single Training Vector: $\mathbf{x}$

- It is worth noting the $\mathbf{x}^{(i)}$ defined above is formally defined to be the i-th column of $\mathbf{X}$, which is the i-th training sample, represented as a $n \times 1$ **column vector**. However, the way we define the Design Matrix is that each row of $\mathbf{X}$ is the transpose of $\mathbf{x}^{(i)}$.
- Note $x_j^{(i)}$ is the value of feature/attribute j in the ith training instance.

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ x_3^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix}_{n \times 1}$$

Target/Label: Y

- This is the target vector. By default, it is a column vector of size $m \times 1$.

$$\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}_{m \times 1}$$

Hypothesis Set: $\mathcal{H}$

- The set where it contains all possible functions to approximate our true function f. Note that the Hypothesis Set can be either continuous or discrete, means to say it can be either a finite or infinite set. But in reality, it is almost always infinite.

Hypothesis: $h : \mathbf{X} \to \mathbf{Y}$ where $\mathbf{x} \mapsto \mathbf{y}$

- Note that this $h \in \mathcal{H}$ is the hypothesis function,
- The final best hypothesis function is called g, which approximates the true function f.

Learning Algorithm: $\mathcal{A}$

- **What this does is from the set of Hypothesis $\mathcal{H}$, the learning algorithm's role is to pick one $h \in \mathcal{H}$ such that this h is the hypothesis function.**
- More often, we also call our final hypothesis learned from $\mathcal{A}$ g.

Hypothesis Subscript $\mathcal{D}$: $h_{\mathcal{D}}$

- This is no different from the previous hypothesis, instead the previous h is a shorthand for this notation.
- This means that the hypothesis we choose is dependent on the sample data given to us, that is to say, given a $\mathcal{D}$, we will use $\mathcal{A}$ to learn a $h_{\mathcal{D}}$ from $\mathcal{H}$.

Generalization Error/Test Error/Out-of-Sample Error: $\mathcal{E}_{\text{out}}(h)$

- **Reference from Foundations of Machine Learning**.
- Given a hypothesis $h \in \mathcal{H}$, a true function $f \in \mathcal{C}$, and an underlying distribution $\mathcal{P}$, the testout-of-sample error of h is defined by

$$\mathcal{E}_{\text{out}}(\mathbf{h}) = \mathbb{P}_{\mathbf{x} \sim \mathcal{P}}[h(\mathbf{x}) \neq f(\mathbf{x})]$$

- Note that the above equation is just the error rate between the hypothesis function h and the true function f and as a result, the test error of a hypothesis is not known because both the distribution $\mathcal{P}$ and the true function f are unknown.
- This brings us to the next best thing we can measure, the In-sample/Empirical/Training Error.

More formally, in a regression setting where we Mean Squared Error,

$$\mathcal{E}_{\text{out}}(\mathbf{h}) = \mathbb{E}_{\mathbf{x}}\left[(h_{\mathcal{D}}(\mathbf{x}) - f(\mathbf{x}))^2\right]$$

This is difficult and confusing to understand. To water down the formal definition, it is worth taking an example, in $\mathcal{E}_{\text{out}}(\mathbf{h})$ we are only talking about the **Expected Test Error** over the Test Set and nothing else. **Think of a test set with only one query point**, we call it $\mathbf{x}_q$, then the above equation is just

$$\mathcal{E}_{\text{out}}(\mathbf{h}) = \mathbb{E}_{\mathbf{x}_q}\left[(h_{\mathcal{D}}(\mathbf{x}_q) - f(\mathbf{x}_q))^2\right]$$

over a single point over the distribution $\mathbf{x}_q$. That is if $\mathbf{x}_q = 3$ and $h_{\mathcal{D}}(\mathbf{x}_q) = 2$ and $f(\mathbf{x}_q) = 5$, then $(h_{\mathcal{D}}(\mathbf{x}_q) - f(\mathbf{x}_q))^2 = 9$ and it follows that

$$\mathcal{E}_{\text{out}}(\mathbf{h}) = \mathbb{E}_{\mathbf{x}_q}\left[(h_{\mathcal{D}}(\mathbf{x}_q) - f(\mathbf{x}_q))^2\right] = \mathbb{E}_{\mathbf{x}_q}[9] = \frac{9}{1} = 9$$

Note that I purposely denoted the denominator to be 1 because we have only 1 test point, if we were to have 2 test point, say $\mathbf{x} = [x_p, x_q] = [3, 6]$, then if $h_{\mathcal{D}}(x_p) = 4$ and $f(x_p) = 6$, then our $(h_{\mathcal{D}}(\mathbf{x}_q) - f(\mathbf{x}_q))^2 = 4$.

Then our

$$\mathcal{E}_{\text{out}}(\mathbf{h}) = \mathbb{E}_{\mathbf{x}}\left[(h_{\mathcal{D}}(\mathbf{x}) - f(\mathbf{x}))^2\right] = \mathbb{E}_{\mathbf{x}}[9, 4] = \frac{1}{2}(9+4) = 6.5$$

Note how I secretly removed the subscript in $\mathbf{x}$, and how when there are two points, we are taking expectation over the 2 points. So if we have m test points, then the expectation is taken over all the test points.

Till now, our hypothesis h is fixed over a particular sample set $\mathcal{D}$. We will now move on to the next concept on **Expected Generalization Error** (adding a word Expected in front makes a lot of difference).

Expected Generalization Error/Test Error/Out-of-Sample Error: $\mathbb{E}_{\mathcal{D}}[\mathcal{E}_{\text{out}}(h)]$

For the previous generalization error, we are only talking a fixed hypothesis generated by one particular $\mathcal{D}$. In order to remove this dependency, we can simply take the expectation of Generalization Error of h over a particular $\mathcal{D}$ by simply taking the expectation over all such $\mathcal{D}_i, i = 1, 2, 3, \ldots K$.

Then the **Expected Generalization Test Error** is independent of any particular realization of $\mathcal{D}$.

$$\mathbb{E}_{\mathcal{D}}[\mathcal{E}_{\text{out}}(h)] = \mathbb{E}_{\mathcal{D}}[\mathbb{E}_{\mathbf{x}}\left[(h_{\mathcal{D}}(\mathbf{x}) - f(\mathbf{x}))^2\right]]$$

In the following example, we can calculate the Expected Generalization Error, where we are using the Error to be Mean Squared Error, so in essence, we are finding the expected MSE.

Empirical Error/Training Error/In-Sample Error: $\mathcal{E}_{\text{in}}(h)$

- Given a hypothesis $h \in \mathcal{H}$, a true function $f \in \mathcal{C}$, and an underlying distribution $\mathcal{P}$, and a sample $\mathbf{X}$ drawn from $\mathcal{X}$ i.i.d with distribution $\mathcal{P}$, the testout-of-sample error of h is defined by

$$\mathcal{E}_{\text{in}}(h) = \frac{1}{m}\sum_{i=1}^{m} \text{sign}(h(\mathbf{x}^{(i)}) \neq f(\mathbf{x}^{(i)}))$$

- Here the sign function is mainly used for binary classification, where if h and f disagrees at any point $x^{(i)}$, then $\text{sign}(h(\mathbf{x}^{(i)}) \neq f(\mathbf{x}^{(i)}))$ evaluates to 1. We take the sum of all disagreements and divide by the total number of samples. In short, that is just the misclassification/error rate.
- The empirical error of $h \in \mathcal{H}$ is average error over the sample, $\mathcal{X}$, in contrast, the generalization error is its expected error based on the distribution $\mathcal{P}$.
- Take careful note here that $h(x^{(i)})$ is the prediction made by our hypothesis (model), we can conventionally call it $\hat{\mathbf{y}}^{(i)}$ whereby our $f(x^{(i)})$ is our ground truth label $y^{(i)}$. I believe that this ground truth label is realized once we draw the sample from $\mathcal{X}$ even though we do not know what f is.
- An additional note here, is that the summand of the in-sample error function is not fixated to the sign function. In fact, I believe you can define any loss function to calculate the "error". As an example, if we are dealing with regression, then we can modify the summand to our favourite Mean Squared Error.

$$\mathcal{E}_{\text{in}}(h) = \frac{1}{m}\sum_{i=1}^{m} h(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(i)})^2$$

Average/Mean Hypothesis: $\bar{h}_{\mathcal{D}} = \mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(\mathbf{x})]$

- This may seem confusing at first sight, but it cactually means the following:
  - Generate many data sets $\mathcal{D}_i, i = 1, 2, 3, \ldots, K$ from population $\mathcal{X} \times \mathcal{Y}$ over a probability distribution $\mathcal{P}$.
  - Apply the learning algorithm $\mathcal{A}$ to each $\mathcal{D}_i$ to get K number of hypothesis $h_i = h_{\mathcal{D}_i}$
  - We can then call the average function for any $\mathbf{x}$ by

$$\bar{h}_{\mathcal{D}} = \mathbb{E}_{\mathcal{D}}[h_{\mathcal{D}}(\mathbf{x})]$$

  - As an example, if there are 3 h_i, where $h_1 = 2x + 1$, $h_2 = 2x + 3$, and $h_3 = 3x + 1$, then given any query point say $x_q = 3$, then $\bar{h}(\mathbf{x}_q) = \frac{h_1 + h_2 + h_3}{3} = \frac{26}{3}$.
- This function, or average hypothesis is vital for our calculation of bias and variance.

Bias - Variance Decomposition

- This is a decomposition of the **Expected** Generalization Error. Formal Proof please read Learning From Data.
- Unless otherwise stated, we consider only the univariate case where $\mathbf{x}$ is a single test point.

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[\mathcal{E}_{\text{out}}(h)] &= \mathbb{E}_{\mathcal{D}}[\mathbb{E}_{\mathbf{x}}\left[(h_{\mathcal{D}}(\mathbf{x}) - f(\mathbf{x}))^2\right]] \\ &= \left(\mathbb{E}_{\mathcal{D}}[\, h_{\mathcal{D}}(\boldsymbol{x})\,] - f(\boldsymbol{x})\right)^2 + \mathbb{E}_{\mathcal{D}}\left[\,\left(h_{\mathcal{D}}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\, h_{\mathcal{D}}(\boldsymbol{x})\,]\right)^2\right] + \mathbb{E}[(\boldsymbol{y} - f(\boldsymbol{x}))^2] \\ &= \left(\bar{h}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 + \mathbb{E}_{\mathcal{D}}\left[\,\left(h_{\mathcal{D}}(\boldsymbol{x}) - \bar{h}(\boldsymbol{x})\right)^2\right] + \mathbb{E}[(\boldsymbol{y} - f(\boldsymbol{x}))^2] \end{aligned}$$

Where $\left(\mathbb{E}_{\mathcal{D}}[\, h_{\mathcal{D}}(\boldsymbol{x})\,] - f(\boldsymbol{x})\right)^2$ is the Bias, $\mathbb{E}_{\mathcal{D}}\left[\,\left(h_{\mathcal{D}}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\, h_{\mathcal{D}}(\boldsymbol{x})\,]\right)^2\right]$ is the Variance and $\mathbb{E}[(\boldsymbol{y} - f(\boldsymbol{x}))^2]$ is the irreducible error ϵ.

Bias: $\left(\mathbb{E}_{\mathcal{D}}[\, h_{\mathcal{D}}(\boldsymbol{x})\,] - f(\boldsymbol{x})\right)^2$

In other form, we can express Bias as

$$\left(\mathbb{E}_{\mathcal{D}}[\, h_{\mathcal{D}}(\boldsymbol{x})\,] - f(\boldsymbol{x})\right)^2 = \left(\bar{h}(\mathbf{x}) - f(\boldsymbol{x})\right)^2$$

See simulation on Bias-Variance Tradeoff to understand.

If our test point is $x_q = 0.9$, then our bias is as such:

$$\widehat{\text{bias}}\left(\hat{f}(0.90)\right) = \frac{1}{n_{\text{value}}}\sum_{i=1}^{n_{\text{value}}}\left(\hat{f}_i^{\,v}(0.90)\right) - f(0.90)$$

Variance: $\mathbb{E}_{\mathcal{D}}\left[\,\left(\,h_{\mathcal{D}}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\, h_{\mathcal{D}}(\boldsymbol{x})\,]\right)^2\,\right]$

This is more confusing, but we first express Variance as:

$$\mathbb{E}_{\mathcal{D}}\left[\,\left(\,h_{\mathcal{D}}(\boldsymbol{x}) - \mathbb{E}_{\mathcal{D}}[\, h_{\mathcal{D}}(\boldsymbol{x})\,]\right)^2\right] = \mathbb{E}_{\mathcal{D}}\left[\,\left(\,h_{\mathcal{D}}(\boldsymbol{x}) - \bar{h}(\mathbf{x})\right)^2\right]$$

If our test point is $x_q = 0.9$, then our variance is as such:

$$\widehat{\text{var}}\left(\hat{f}(0.90)\right) = \frac{1}{n_{\text{value}}}\sum_{i=1}^{n_{\text{value}}}\left(\hat{f}_i^{\,v}(0.90) - \frac{1}{n_{\text{value}}}\sum_{i=1}^{n_{\text{value}}}\hat{f}_i^{\,v}(0.90)\right)^2$$

- **Loss Function:** $\mathcal{L}$

- **Performance Metric:** $\mathcal{M}$

Pseudo Code

Cross-Validation

- Define G as the set of combination of hyperparamters. Define number of splits to be K.
- For each set of hyperparameter $z \in Z$:
  - for fold j in K:
    - Set $F_{\text{train}} = \bigcup_{\substack{i=1 \\ i \neq j}}^{K} F_i$
    - Set $F_{\text{val}} = F_j$ as the validation set
    - Perform Standard Scaling on F_{train} and find the mean and std
    - Perform VIF recursively on F_{train} and find the selected features
    - Transform F_{val} using the mean and std found using F_{train}
    - Transform F_{val} to have only the selected features from F_{train}
    - Train and fit on F_{train}
  - Evaluate the fitted parameters on F_{val} to obtain $\mathcal{M}$

Readings and References

- False-positive and false-negative cases of fine-needle aspiration cytology for palpable breast lesions
- What is a Dendrogram?
- Breast Biopsy - Mayo Clinic
- Data Centric - Andrew Ng
- When is Multicollinearity not an issue - Paul Allison
- Intuitive Explanation of Multicollinearity in Linear Regression - Stackoverflow
- Hypothesis Testing Across Models
- Hypothesis Test for Comparing ML Algorithms - Jason Brownlee
- Regression Modelling Strategies - Professor Frank Harrell
- Damage Caused by Classification Accuracy and Other Discontinuous Improper Accuracy Scoring Rules - Professor Frank Harrell
- On a reliable cross validation split 1
- On a reliable cross validation split 2
- Estimate Generalization Error
- Using Bootplot to compare Model's Performance