6.1 Training and Testing Errors

Let's consider the multiple linear regression model with p predictors **plus** the intercept, i.e.

$$Y \sim X_1 + X_2 + \ldots + X_p$$

In many applications, the number of explanatory variables, i.e., p is large and in some cases we could even have $n \ll p$. But, this does not necessarily mean that all the variables are relevant to the response Y. In fact, *only a small portion* of the p variables are believed to be relevant to Y.

Our **goal** in this chapter is to develop methods that will allow us to efficiently identify the set of predictors that are useful estimating/predicting the response. That is, we need to identify

$$S = \{j : \beta_j \neq 0\}$$

So far in this course, we have discussed the importance of creating a model that is good for estimation purposes, that satisfies all model assumptions and includes variables that are statistically significant. We also mentioned that when our main purpose is to build a strong predictive model, we can allow our model to deviate from some of the assumptions. So, if our task is to go well on prediction, then *why is it important to remove unnecessary variables from the model*?

Recall that the least squares estimator $\hat{\beta}$ is unbiased, which means that estimators for irrelevant $\hat{\beta}_j$ (with $j \in S^c$) will eventually go to zero anyway.

To better understand the implications of unnecessary parameters in a MLR model, let us further discuss and quantify the **Training** and **Testing Errors**.

Consider that we *split our data in two parts*:

- **Training data**: $(\mathbf{x}_i, y_i)_{i=1}^n$ used to fit our model
- **Testing data**: $(\mathbf{x}_i, y_i^*)_{i=1}^n$ an **independent** data set collected at the same locations $\mathbf{x}_i$ (also known as in-sample prediction)

Remark: In practice, we are given a full data set to analyze, which we then need to split it in two parts (typically in a random fashion) - a *training* part and a *testing* part with a higher percentage of data in the training (usually >70%).

Since both *testing* and *training* data come from the same population or are collected at the same locations $\mathbf{x}_i$, statistically we can write both models as follows:

$$\mathbf{y}_{n\times 1},\ \mathbf{y}^*_{n\times 1} \sim^{iid} N_n(\mu, \sigma^2\mathbf{I}_n) \text{ and } \mu = \mathbf{X}\beta$$

We can also write:

$$\begin{aligned}\mathbf{y} &= \mathbf{X}\beta + \varepsilon \\ \mathbf{y}^* &= \mathbf{X}\beta + \varepsilon^*\end{aligned}$$

with $\varepsilon_{n\times 1} \sim^{iid} \mathcal{N}_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$, $\varepsilon^*_{n\times 1} \sim^{iid} \mathcal{N}_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$ independent.

Having these models in mind, we compute the MSE for train and testing errors

$$\begin{aligned}\mathbb{E}(\text{Train Error})^2 &= \mathbb{E}||\mathbf{y} - \hat{\mathbf{y}}||^2 = \mathbb{E}||(\mathbf{I}-\mathbf{H})\mathbf{y}||^2 \\ &= tr((\mathbf{I}-\mathbf{H})Cov(\mathbf{y})(\mathbf{I}-\mathbf{H})^\top) \\ &= \sigma^2 tr((\mathbf{I}-\mathbf{H})) = (n-p)\sigma^2 \\ &= n\sigma^2 - p\sigma^2\end{aligned}$$

$$\begin{aligned}\mathbb{E}(\text{Test Error})^2 &= \mathbb{E}||\mathbf{y}^* - \mathbf{X}\hat{\beta}||^2 \\ &= \mathbb{E}||(\mathbf{y}^* - \mathbf{X}\beta) + (\mathbf{X}\beta - \mathbf{X}\hat{\beta})||^2 \\ &= \mathbb{E}||\mathbf{y}^* - \mu||^2 + \mathbb{E}||\mathbf{X}\beta - \mathbf{X}\hat{\beta}||^2 \\ &= \mathbb{E}||\varepsilon^*||^2 + tr(\mathbf{X}Cov(\hat{\beta})\mathbf{X}^\top) \\ &= n\sigma^2 + \sigma^2 tr\mathbf{H} = n\sigma^2 + p\sigma^2\end{aligned}$$

From the previous equations we can conclude that:

- the *training error decreases with* p.
- the *testing error increases with* p.

This implies that if our goal is *pure prediction*, adding more variables to matrix $\mathbf{X}$ is not the best option. But, *does this imply that the intercept-only model with $p = 0$, i.e. the one with the smallest expected test error is the best*?

No! The previous analysis is based on the assumption that the mean of $\mathbf{y}$ is in the column space of $\mathbf{X}$, i.e., there exists some coefficient vector β such that $\mu = \mathbf{X}\beta$. In general, we run a linear regression model using only a *subset* of the columns of $\mathbf{X}$. This means there will be *an additional **Bias** term*.

MSE of Training and Testin Errors when $\mathbf{X}_\gamma$ is used

Index *each* model (i.e., each subset of the p variables) by a p-dimensional binary vector γ:

$$\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_p), \quad \text{where } \gamma_j = \begin{cases} 1, \text{if } X_j \text{ is included in the model} \\ 0, \text{ otherwise} \end{cases}$$

For example, $\gamma = (1, 1, \ldots, 1)$ refers to the *full* model including all p variables, while $\gamma = (0, 0, \ldots, 0)$ refers to the *intercept-only* model. Based on the different combinations of p variables in and out of the model, there are a total of 2^p possible subsets or sub-models.

Now, we will quantify the training and testing errors for a sub-model with design matrix $\mathbf{X}_\gamma$. We assume that we fit the data $\mathbf{y}$ with respect to a linear model with a sub-design matrix $\mathbf{X}_\gamma$ where $\mathbf{X}_\gamma$ contains only columns from $\mathbf{X}$ such that $\gamma_j = 1$.

$$\mathbb{E}(\text{Test Error})^2 = n\sigma^2 + p\sigma^2 + Bias_\gamma$$
$$\mathbb{E}(\text{Training Error})^2 = n\sigma^2 - p\sigma^2 + Bias_\gamma$$

- Bigger model (i.e., p large) $\rightarrow$ small Bias, but large Variance ($p\sigma^2$)
- Smaller model (i.e., p small) $\rightarrow$ large Bias, but small Variance ($p\sigma^2$).

To reduce the *test error* (i.e., prediction error), the key is to find the best trade-off between Bias and Variance.

To do that, we have two types of methods:

- Testing-based: Select best model based on statistical tests for model comparison.

- Criterion-based: Select best model based on information criteria (combining model fit and model complexity) for model comparison.