Machine Learning Model Evaluation

Overfitting-Underfitting & Bias-Vairiance TradeOff

Dr. Saad Laouadi

Econometrics and Data Science Academy

Important Primary Lecture in ML Model Evaluation, November 2021

Model Assessment

It is crucial to understand that in statistical learning methods:

- No method dominates all others over all possible data-sets.
- In each problem, a method can outperform all others.
- In the same problem, but different a data set, A statistical method can outperform the other.

For this reason:

- A data analyst must try different methods.
- Assess each method.
- Decide on which method they will proceed with based on its **performance**.

Statistical Analysis: is a challenging domain, not just based on statistical learning methods, but on many characteristics:

- **The mind**: we understand problems differently.
- **Computing skills**: How tweak the elements of a specific algorithm to get the best out of it.
- **Expertise**: Selecting the convenient algorithms for certain project.
- **Analytic skills**: Feature engineering, features selection . . .

Measuring the Performance of Algorithms

Model evaluation is based on its predictions on new (unseen) data, which we call **test data**.

1- **MEAN SQUARED ERROR**: This measure in commonly used in regression.

$$\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$$

- If the MSE is calculated on the train data we call it Train MSE
- If the MSE is calculated on the test data we call it Test MSE, which is the one we are interested in.
- We would like to have as small the **Test MSE** as possible.
- If we have different algorithms, we would pick up the one with smallest **Test MSE**.

Overfitting VS. Underfiting

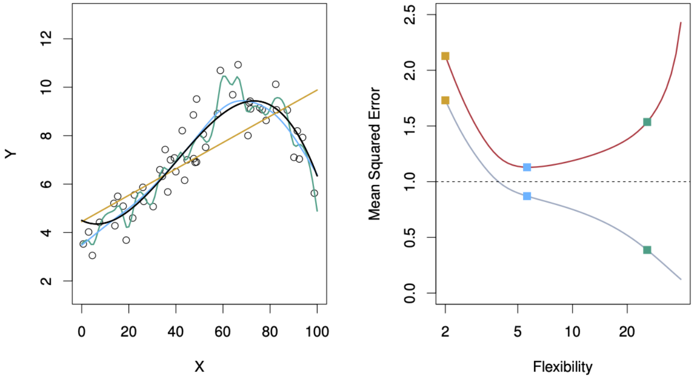
- **Overfitting**: Where the algorithm fits the noise or **patterns that are happened by random chance** in the training data, it other words **tracks almost every point**. Overfitting is known to have small Train MSE and Large Test MSE.
- **Underfitting**: The algorithm is not flexible enough to catch the true form of the data. In this case we have large Train MSE and Large Test MSE.

Important in Machine Learning

Generally : we are after a situation where we neither **overfit** nor **underfit** (most of the work is done here trying to find the best algorithm that fits this situation.)

- If a model does not **overfit**, we say it **generalizes well**

Overfitting VS. Underfiting Example 01



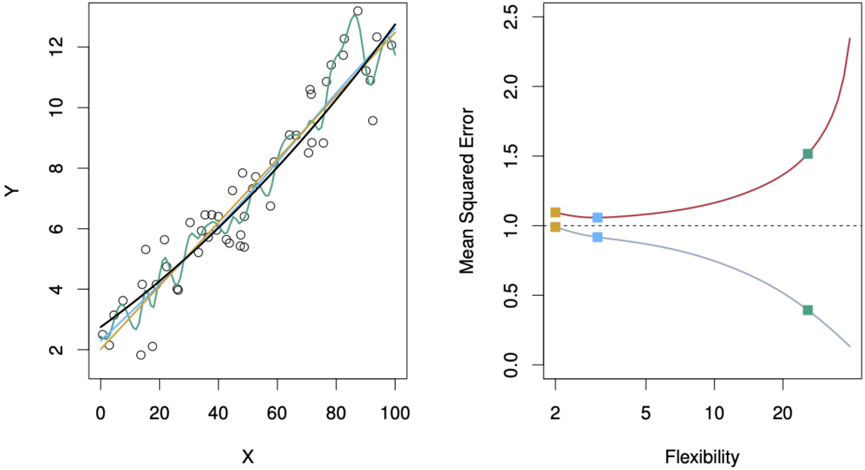
Left: **Orange**: linear Regression, **Blue and Green**: smoothing spline. **Black**: the true function form. right: **Gray curve**: Train MSE, **Red curve**: Test MSE

Insights about overfitting

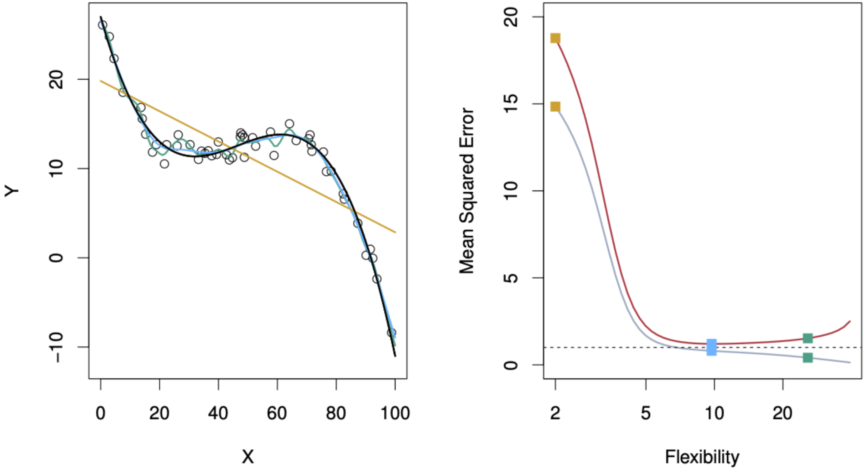
1. Linear Regression is inflexible
2. The more flexible the function the more it fits the observations closely. which is too **wigly**
3. Train MSE is always less than Test MSE
4. Train MSE declines as flexibility increases
5. Test MSE declines as flexibility increases but at some point it levels off and then starts to increase (**This is a sign of overfitting**.) This is known as a **U-shape in Test MSE**
6. The wiggly curve has the smallest **Train MSE** but the worst Test MSE as well as linear regression.
7. Linear Regression **orange** shows underfitting.
8. The wiggly curve **the green** shows overfitting.
9. The blue curve is the closest one to the true form, in this case it is the best fit.

Result: The best function can be any function like: Logistic Regression, Random Forest, or Neural Network ...

Overfitting VS. Underfiting Example 02



Overfitting VS. Underfiting Example 03



The Bias-Variance Trade-Off (Mathematically)

The Expected value of Test MSE is composed of the components (variance and bias plus the variance error), sometimes it is called **a generalization error**, the formula is shown below:

$$\mathbf{E}(\mathbf{MSE})^2 = \mathbf{Var}\left(\widehat{\mathbf{y}}\right) + \left[\mathbf{Bias}\left(\widehat{\mathbf{y}}\right)\right]^2 + Var\left(\varepsilon\right)$$

$$\mathbf{MSE} = \left(\mathbf{y} - \widehat{\mathbf{y}}\right)$$

- This quantity is the **expected Test Mean Squared Error**.

Bias-Variance tradeoff (Continue)

1. **The expected MSE** refers to the average test **MSE** from repeatedly estimated function using a large number of training sets, then tested on test sets.
2. The previous formula tells us that we need to find an algorithm that minimizes the **test (MSE)**. That can happen when we have both **low variance and low bias**
3. This formula is always positive, the variance is positive plus a positive value of the bias.
4. The relationship between **Test MSE, variance and Bias** is referred to as the bias-variance trade-Off

The Bias-Variance Trade-Off: (Literally)

Variance

1. **Variance**: refers to the amount by which a **fitted function** would change if we estimated it using a **different dataset**.

- We should not have a function that varies too much between training sets.
- If a statistical method (say Decision Tree) has a high variance, fitting the same method or algorithm on a different training set would result in totally different results.
- More flexible algorithms usually have higher variance like **decision trees**

Examples

1. in figure 01: the green curve has **high variance** because it is too flexible. If we change only few points, the fitted function would change considerably.
2. The linear regression (orange in figure 1) has **low variance**.

The Bias-Variance Trade-Off: (Literally)

Bias

2-**Bias**: Refers to the error that is introduced by approximating a **real-life** problem. In other words: if the real form of a function is quadratic but we fit a linear regression, thus, we have made an error, precisely we would always have biased results.

- Generally, more flexible algorithms have low-bias

Example

In figure 03: The true form of the function is non-linear. Thus, no matter how many training observations we are given, it will not be possible to produce an accurate estimate using **Linear Regression**. In this Case, Linear Regression has **high-bias**

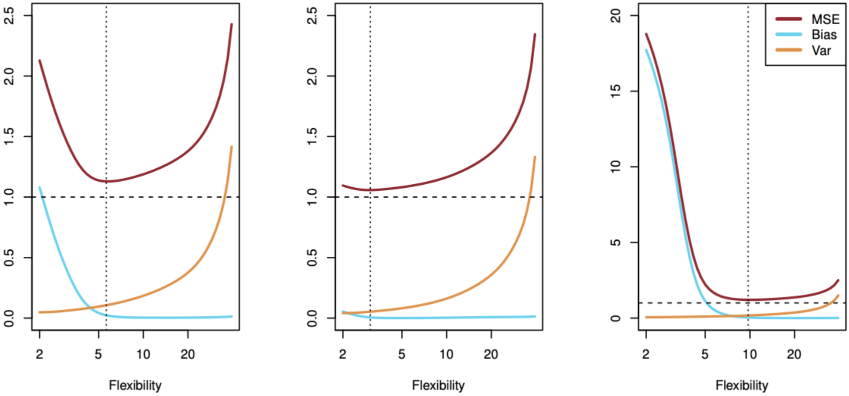
***In figure 02*: The linear regression would be a good fit.**

General Thoughts about Variance and Bias

- Flexible algorithms have **high variance** but **low bias**
- The change in **variance** and **bias** can be captured using **MSE** (formula in the previous slide).
- Increasing the flexibility of a method would result in fast decrease in bias more than the increase in **variance**. We see Test MSE declining
- At some point increasing flexibility will have less effect on **bias** but results in a significant increase in **variance**. The Test MSE will start increasing.
- We seek trading between increasing flexibility to a certain point where we have **low variance and low-bias**.

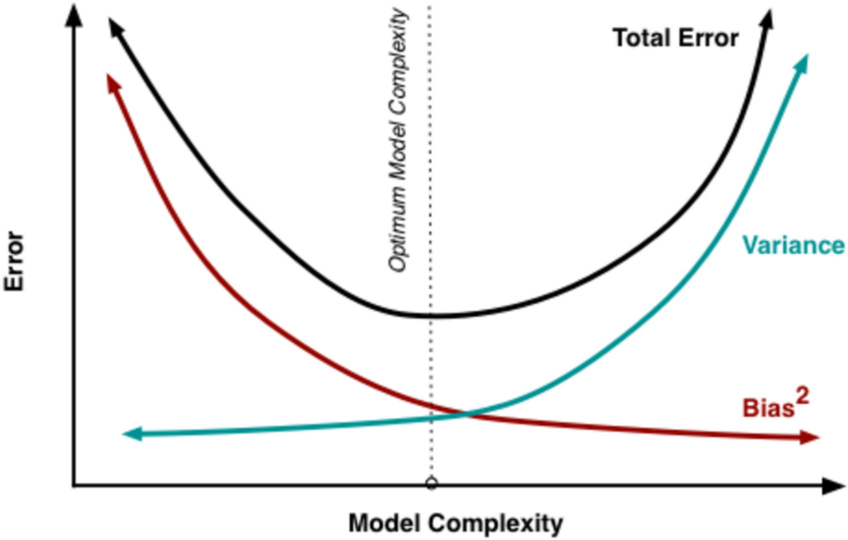
Bias-Variance Trade-Off (Graphically)

Example 01



Bias-Variance Trade-Off (Graphically)

Example 02



How Bias-Variance is conducted in practice

Of course, the true form of the model is unknown, for this reason we follow the next steps:

1. Collecting the data needed for the specific problem
2. Splitting the data into train/test (also called **hold-out set**) (sometimes the data is split into three parts Train/Validation and test)
3. Fitting the data on train set
4. evaluate the model on the test set or
5. Using **Cross Validation Technique** which is very effective in practice.

Remedies

- In case of overfitting: Decrease the flexibility.
- In case of underfitting: Increase the flexibility, or gather more features or data points ....