Machine Learning

Bias-Variance Tradeoff

Alberto Maria Metelli

Credits to Francesco Trovò

Bias-Variance Dilemma

Known Process

To explicitly analyse the variance and the bias of a model we need to know the process generating the data:

$$t = \underbrace{f(x)}_{\text{deterministic}} + \underbrace{\varepsilon}_{\text{noise}} \qquad\qquad f(x) = 1 + \frac{1}{2}x + \frac{1}{10}x^2$$

- the input are x **uniformly** distributed in $[0, 5]$, i.e., $p(x) = \mathrm{Uni}([0, 5])$
- the noise ε distribution $p(t|x)$ has $\mathbb{E}[\varepsilon] = 0$ and $Var(\varepsilon) = \sigma^2 = 0.7^2$

Two-Model Dilemma

Assume to approach the learning problem (we do not know the true model) using either one of the two following models:

$$
\begin{aligned}
\mathcal{H}_1 : \quad & y(x) = a + bx \\
\mathcal{H}_2 : \quad & y(x) = a + bx + cx^2
\end{aligned}
$$

Population Risk Minimization

- Hypothesis space: $y(x) \in \mathcal{H}$
- Loss function: squared loss function $(t - y(x))^2$
- We know $p(x, t)$
- **Population** risk minimization:

$$y^* \in \arg\min_{y \in \mathcal{H}} \mathbb{E}[(t - y(x))^2] = \int p(x, t)(t - y(x))^2 dxdt$$

We can solve this problem only if we know $p(x, t)$!

Population Risk Minimization

If the real model is known we can compute the optimal model for the two hypothesis space:

$$\mathcal{H}_1: \qquad \arg\min_{(a,b)} \int_0^5 \frac{1}{5}(f(x) - a - bx)^2 dx = \left(\frac{7}{12},\ 1\right)$$

$$\mathcal{H}_2: \qquad \arg\min_{(a,b,c)} \int_0^5 \frac{1}{5}(f(x) - a - bx - cx^2)^2 dx = \left(1,\ \frac{1}{2},\ \frac{1}{10}\right)$$

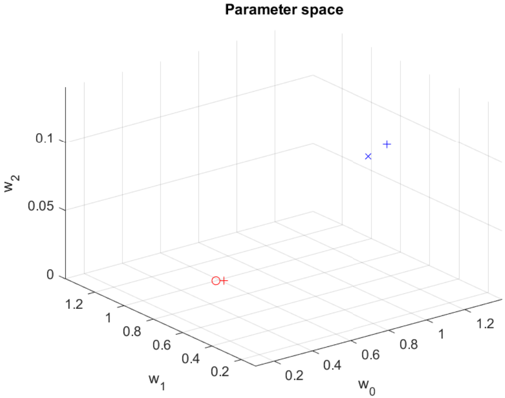
Empirical Risk Minimization

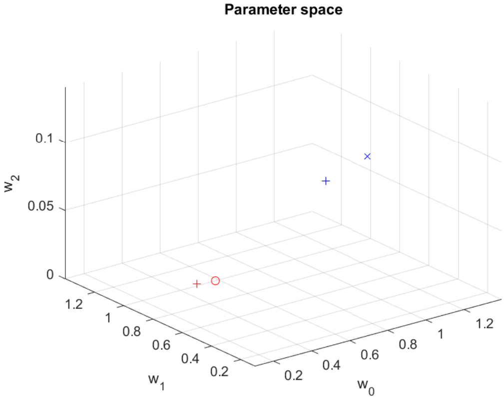
- Hypothesis space: $y(x) \in \mathcal{H}$
- Loss function: squared loss function $(t - y(x))^2$
- We **do not know** $p(x, t)$ but we have samples $\mathcal{D} = \{(x_n, t_n)\}_{n=1}^N$ i.i.d. from p
- **Empirical** risk minimization:

$$\widehat{y} \in \arg\min_{y \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} (t_n - y(x_n))^2$$

$\widehat{y}$ depends on the employed dataset $\mathcal{D}$

Optimal Parameters and Realized Parameters



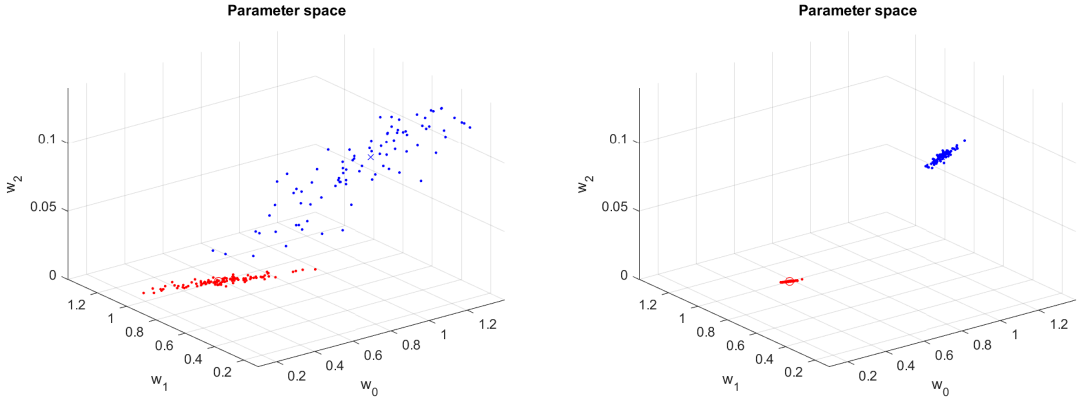


The blue x is the best model in $\mathcal{H}_2$ and the red circle is the best model in $\mathcal{H}_1$

The pluses are the optimal parameters for two realization of the dataset ($N = 1000$)

Visualization of Bias and Variance

If we repeat the process for multiple times (generation of 100 independent dataset) with different number of samples ($N = 100$ on the left and $N = 10000$ on the right)



Computation of Bias and Variance

In this specific case we can even estimate the Bias and Variance of the two models:

$$\mathbb{E}_{\mathcal{D},t}[(t - \widehat{y}(x))^2] = \sigma^2 + Var_{\mathcal{D}}[\widehat{y}(x)] + \mathbb{E}_{\mathcal{D}}[f(x) - \widehat{y}(x)]^2$$

- $t = f(x) + \epsilon$ where $\mathbb{E}[\epsilon] = 0$ and $Var[\epsilon] = \sigma^2$
- Fixed (unseen point) x
- Expectation taken w.r.t. the training dataset $\mathcal{D}$ and t

Computation of Bias and Variance

```
Linear error: 0.46867
Linear bias: 0.03613
Linear variance: 0.00011514
Linear sigma: 0.43242
Quadratic error: 0.42146
Quadratic bias: 1.412e-06
Quadratic variance: 0.00014674
Quadratic sigma: 0.42131
```

All the considerations holds on average, therefore there might be realizations for which the Bias and Variance of different models might not be coherent with what we saw.

Bias-Variance Tradeoff

Model Selection Problem

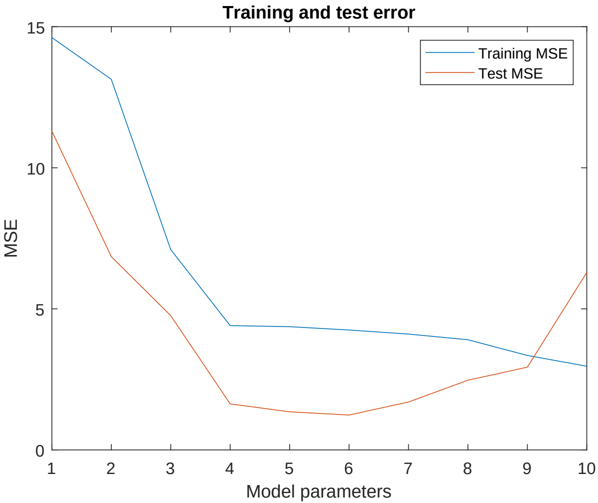
In real scenarios we do not know the real model, so we should select the correct one among a set of models.

Consider the possible solutions for a regression problem:

- Hypothesis space: $y_n = f(x_n, w) = \sum_{k=0}^{o} x_n^k w_k$
- Loss measure: $RSS(w) = \sum_n (y_n - t_n)^2$
- Optimization method: Least Square (LS)

The order o and other parameters which should be chosen before performing the training phase are usually addressed as *hyperparameters*

Limits of Using the Training error

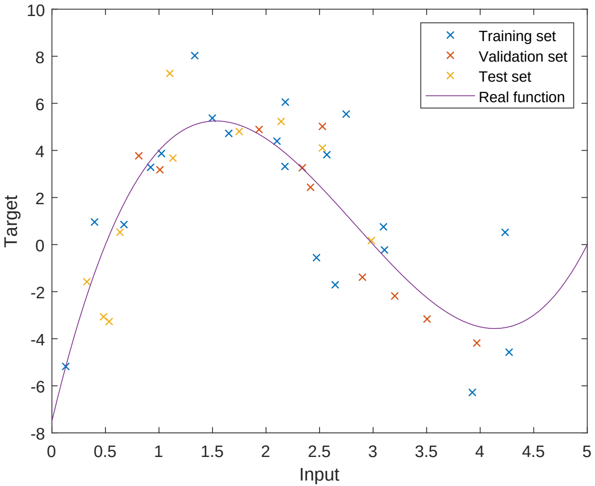


Validation

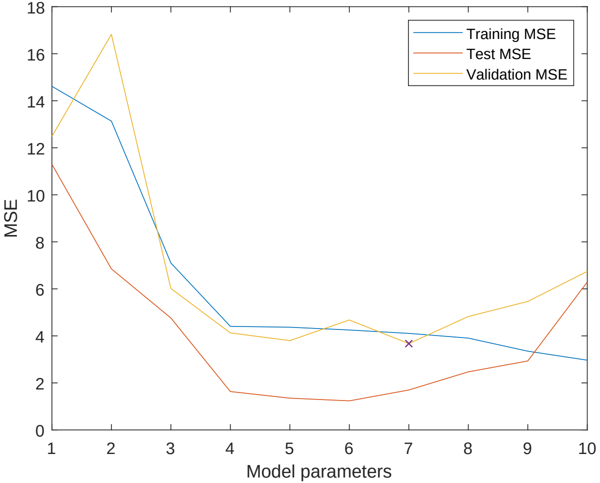
- Training set X_{train}, i.e., the data we will use to **learn** the model parameters
- Validation set X_{vali}, i.e., the data we will use to **select** the model
- Test set X_{test}, i.e., the data we will use to **evaluate** the performance of our model

Usually, we use a split proportional to 50%-25%-25% for the three sets

Dataset Generated



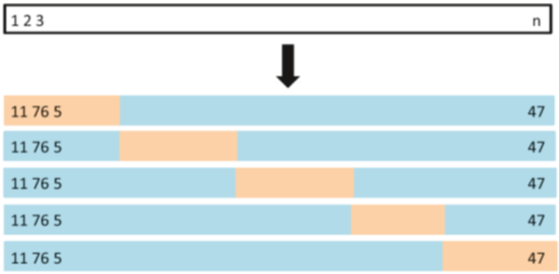
Validation Results



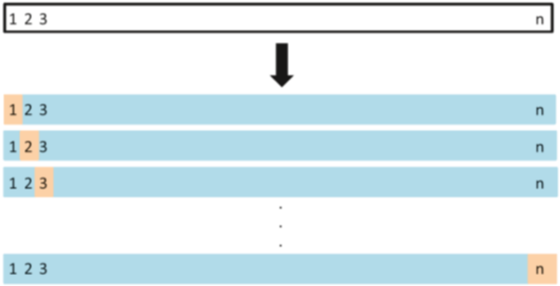
LOO and Crossvalidation

This way we reduce the amount of samples we could use for training of 33%, which could compromise the analysis since the training has been performed with a significantly smaller dataset

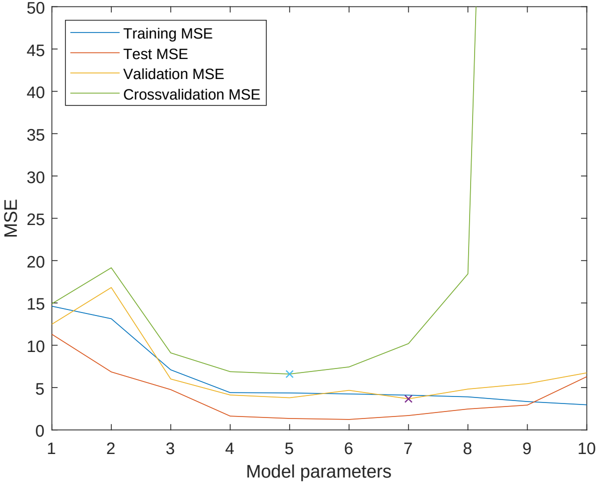
Crossvalidation



Leave One Out



Crossvalidation Results ($K = 5$)



Checking the Results

The data have been generated from the following model:

$$y = (0.5 - x)(5 - x)(x - 3) + \varepsilon$$

where $\varepsilon \sim \mathcal{N}(0, 1.5^2)$

The correct order is then $o = 3$ (4 in the graphs which considers also the constant term)

The procedure is correct on average, the realizations might return different orders than the correct one

Computational Times

Using different methods we have different time for the model selection:

```
Elapsed time is 0.016354 seconds. % Validation
Elapsed time is 0.431666 seconds. % Crossvalidation
Elapsed time is 4.308715 seconds. % LOO
```

Depending on the computational power available and the number of data we have we might choose different methods