Machine Learning Course - CS-433

Bias-Variance Decomposition

Oct 12, 2022

minor changes by Nicolas Flammarion 2021,2020, minor changes by Rüdiger Urbanke 2019, changes by Martin Jaggi 2018, changes by Rüdiger Urbanke 2017 ©Mohammad Emtiyaz Khan and Rüdiger Urbanke 2016

Last updated on: October 9, 2022

EPFL

Motivation

Last time we saw how to assess if a given function was good. In particular, we discussed how we can bound the difference between the true risk of the function and the empirical risk. We then used the same ideas and discussed how to choose the "best" out of a finite number of models. This led us to the idea of splitting the data into a *train* set and a *test* set. Our motivation for the model selection problem was that typically we need to optimize hyper-parameters. E.g., in the ridge regression problem the hyper-parameter was λ. These hyper-parameters often control the "complexity" of the class of models that we allow.

Today we will focus on how the risk (true or empirical) behaves as a function of the complexity of the model class. This will lead to the important concept of the bias - variance trade-off when we perform the model selection. It will help us to decide how "complex" or "rich" we should make our model.

Let us discuss a very simple example. Consider linear regression with a one-dimensional input and using polynomial feature expansion. The maximum degree d regulates the complexity of the class. We will see that the following is typically true.

Assume that we only allow simple models, i.e., we restrain the degree to be small:

- We then typically will get a large bias, i.e., a bad fit.
- On the other hand the variance of $L_{\mathcal{D}}(f_S)$ as a function of the random sample S is typically small.

We say that we have high bias but low variance.
Assume that we allow complex models, i.e., we allow large degrees:

- We then typically will find a model that fits the data very well. We will say that we have small bias.
- But we likely observe that the variance of $L_{\mathcal{D}}(f_S)$ as a function of the random sample S is large.

We say that we have low bias but high variance.

Data Generation Model

Assume that the data is generated as

$$y = f(\mathbf{x}) + \varepsilon,$$

where f is some (arbitrary and unknown) function and ε is additive *noise* with distribution $\mathcal{D}_\varepsilon$ that is independent from sample to sample and independent from the data. Assume the noise has zero mean (otherwise this constant can be absorbed into f). Note that f is in general not *realizable*, i.e., it is in general not in our model class.
We further assume that $\mathbf{x}$ is generated according to some fixed but unknown distribution $\mathcal{D}_{\mathbf{x}}$. Finally, we assume that the loss function $\ell(\cdot, \cdot)$ is the square loss. Let $\mathcal{D}$ denote the joint distribution on pairs $(\mathbf{x}, y)$.

Error Decomposition

As always, we have given some training data S_{train}, consisting of i.i.d. samples according to $\mathcal{D}$. Given our learning

algorithm $\mathcal{A}$, we compute the prediction function $f_{S_{\text{train}}} = \mathcal{A}(S_{\text{train}})$. We are ultimately interested in how the true error

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[(f(\mathbf{x}) + \varepsilon - f_{S_{\text{train}}}(\mathbf{x}))^2]$$

behaves as a function of the training set S_{train} and the complexity of the model class.
But the decomposition we will discuss already applies "pointwise", i.e., for a single sample $\mathbf{x}$. It is therefore simpler if we fix $\mathbf{x}_0$, and only consider

$$(f(\mathbf{x}_0) + \varepsilon - f_{S_{\text{train}}}(\mathbf{x}_0))^2.$$

We imagine that we are running the experiment many times: we create S_{train}, we learn the model $f_{S_{\text{train}}}$, and then we evaluate the performance by computing the square loss for this fixed element $\mathbf{x}_0$.
So let us look at the expected value of this quantity:

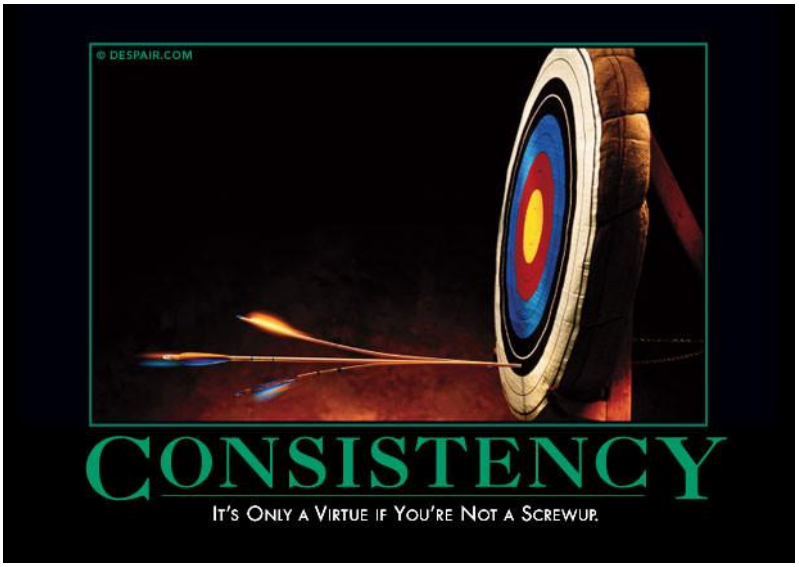
$$\mathbb{E}_{S_{\text{train}}\sim\mathcal{D},\, \varepsilon\sim\mathcal{D}_\varepsilon}[(f(\mathbf{x}_0) + \varepsilon - f_{S_{\text{train}}}(\mathbf{x}_0))^2].$$

We will now show that we can rewrite the above quantity as a sum of *three non-negative terms* and this decomposition

has a natural interpretation. We write

$$
\begin{aligned}
&\mathbb{E}_{S_{\text{train}} \sim \mathcal{D},\, \varepsilon \sim \mathcal{D}_\varepsilon}\big[(f(\mathbf{x}_0) + \varepsilon - f_{S_{\text{train}}}(\mathbf{x}_0))^2\big] \\
&\overset{(a)}{=} \mathbb{E}_{\varepsilon \sim \mathcal{D}_\varepsilon}[\varepsilon^2] + \mathbb{E}_{S_{\text{train}} \sim \mathcal{D}}\big[(f(\mathbf{x}_0) - f_{S_{\text{train}}}(\mathbf{x}_0))^2\big] \\
&\overset{(b)}{=} \mathrm{Var}_{\varepsilon \sim \mathcal{D}_\varepsilon}[\varepsilon] + \mathbb{E}_{S_{\text{train}} \sim \mathcal{D}}\big[(f(\mathbf{x}_0) - f_{S_{\text{train}}}(\mathbf{x}_0))^2\big] \\
&\overset{(c)}{=} \underbrace{\mathrm{Var}_{\varepsilon \sim \mathcal{D}_\varepsilon}[\varepsilon]}_{\text{noise variance}} \\
&\qquad + \underbrace{\big(f(\mathbf{x}_0) - \mathbb{E}_{S'_{\text{train}} \sim \mathcal{D}}[f_{S'_{\text{train}}}(\mathbf{x}_0)]\big)^2}_{\text{bias}} \\
&\qquad + \mathbb{E}_{S_{\text{train}} \sim \mathcal{D}}\Big[\underbrace{\big(\mathbb{E}_{S'_{\text{train}} \sim \mathcal{D}}[f_{S'_{\text{train}}}(\mathbf{x}_0)] - f_{S_{\text{train}}}(\mathbf{x}_0)\big)^2}_{\text{variance}}\Big].
\end{aligned}
$$

Note that here S'_{train} is a second training set, also sampled from $\mathcal{D}$ that is independent of the training set S_{train}.



Details:
In step (a), we omitted the third term

$$\mathbb{E}_{S_{\text{train}}\sim\mathcal{D},\,\varepsilon\sim\mathcal{D}_\varepsilon}\big[2\varepsilon(f(\mathbf{x}_0)-f_{S_{\text{train}}}(\mathbf{x}_0))\big].$$

But since the noise ε is independent from S_{train} we can first average over the noise, and by observing that the noise has mean zero, we see that this term is in fact zero.
Further, since the noise has zero mean, the second moment is equal to the variance. This explains step (b).
In step (c) we have added and subtracted the constant term $\mathbb{E}_{S'_{\text{train}}\sim\mathcal{D}}[f_{S'_{\text{train}}}(\mathbf{x}_0)]$ to the expression and then expanded the square.
The expansion yields the two expressions which are stated (termed "bias" and "variance"). In addition it yields the cross term (to save space we omit the factor 2 and the $_{\text{'train'}}$-subscript)

$$\begin{aligned}
&\mathbb{E}_{S\sim\mathcal{D}}\Big[\big(f(\mathbf{x}_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(\mathbf{x}_0)]\big)\cdot\big(\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(\mathbf{x}_0)]-f_S(\mathbf{x}_0)\big)\Big]\\
&=\big(f(\mathbf{x}_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(\mathbf{x}_0)]\big)\cdot\mathbb{E}_{S\sim\mathcal{D}}\big[(\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(\mathbf{x}_0)]-f_S(\mathbf{x}_0))\big]\\
&=\big(f(\mathbf{x}_0)-\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(\mathbf{x}_0)]\big)\cdot\big(\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(\mathbf{x}_0)]-\mathbb{E}_{S\sim\mathcal{D}}[f_S(\mathbf{x}_0)]\big)\\
&=0.
\end{aligned}$$

Interpretation of Decomposition

Each of the three terms is non-negative. Hence each of them is a lower bound on the true error for the input $\mathbf{x}_0$.
The noise imposes a strict lower bound on what error we can achieve. This contribution is given by the term $\mathrm{Var}_{\varepsilon\sim\mathcal{D}_\varepsilon}[\varepsilon]$.
The bias term is the square of the difference between the actual value $f(\mathbf{x}_0)$ and the expected prediction $\mathbb{E}_{S'\sim\mathcal{D}}[f_{S'}(\mathbf{x}_0)]$, where the expectation is over the training sets. (E.g. simple models can not fit well, so have a large bias)
The variance term is the variance of the prediction function. If we consider very complicated models then small variations in the data set can produce vastly different models and our prediction for an input $\mathbf{x}_0$ will vary widely.

Examples

The following four figures are take from the book by James, Witten, Hastie, and Tibshirani (Introduction to Statistical Learning).
The first three pictures show three different functions each (the true function is the black curve). The first function has medium "complexity", the second is very simple, and the third is the most complicated. In each case, three different predictions are done based on models of increasing complexity.

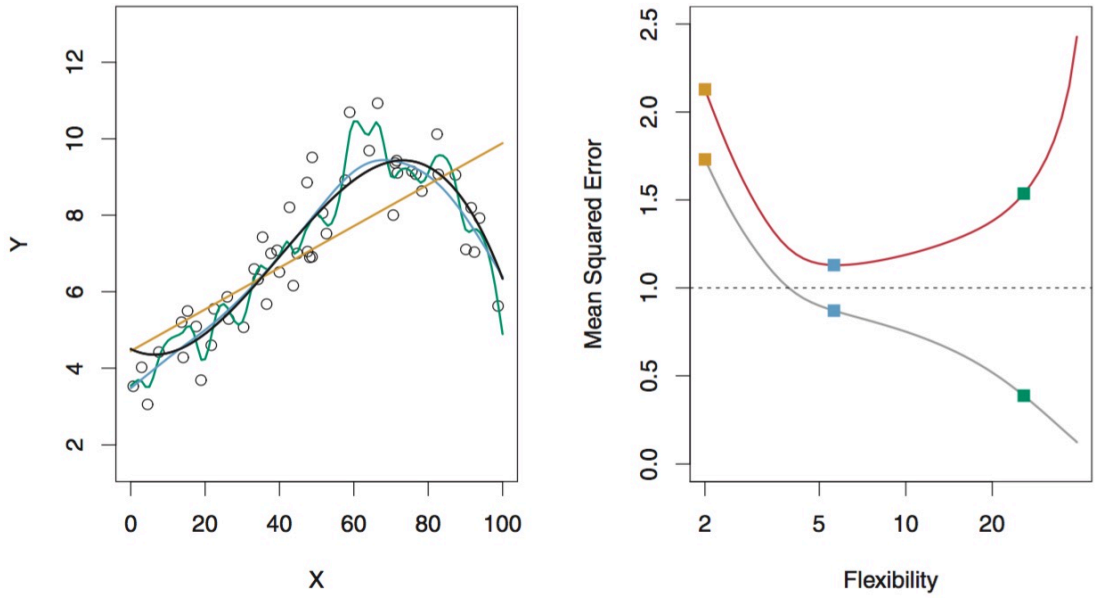


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

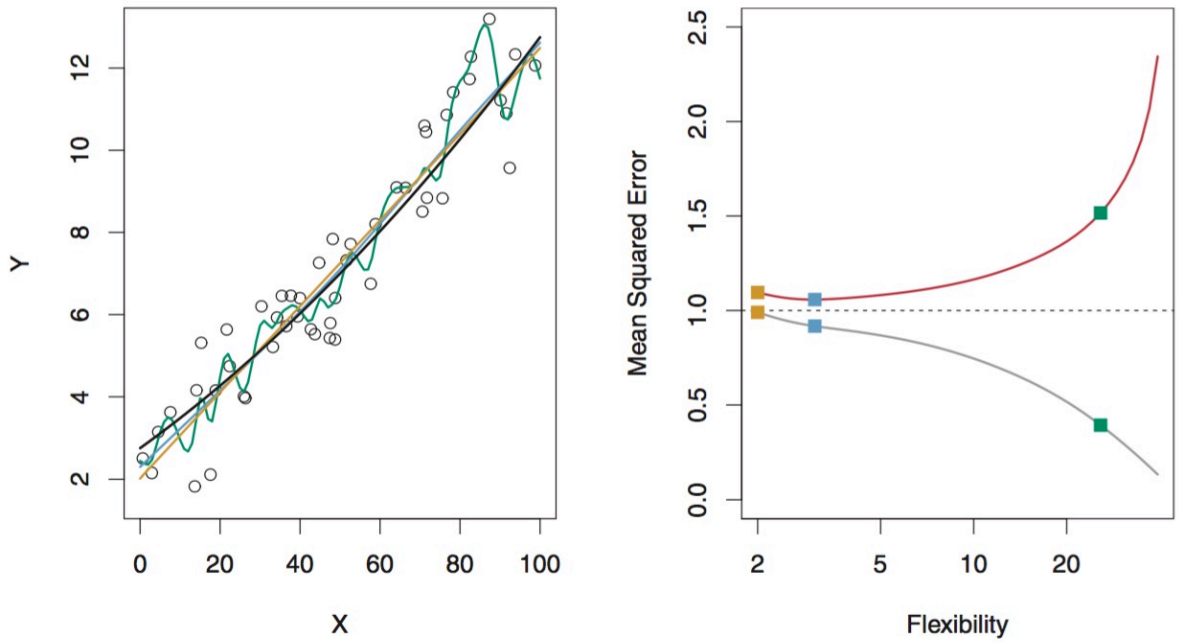


FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

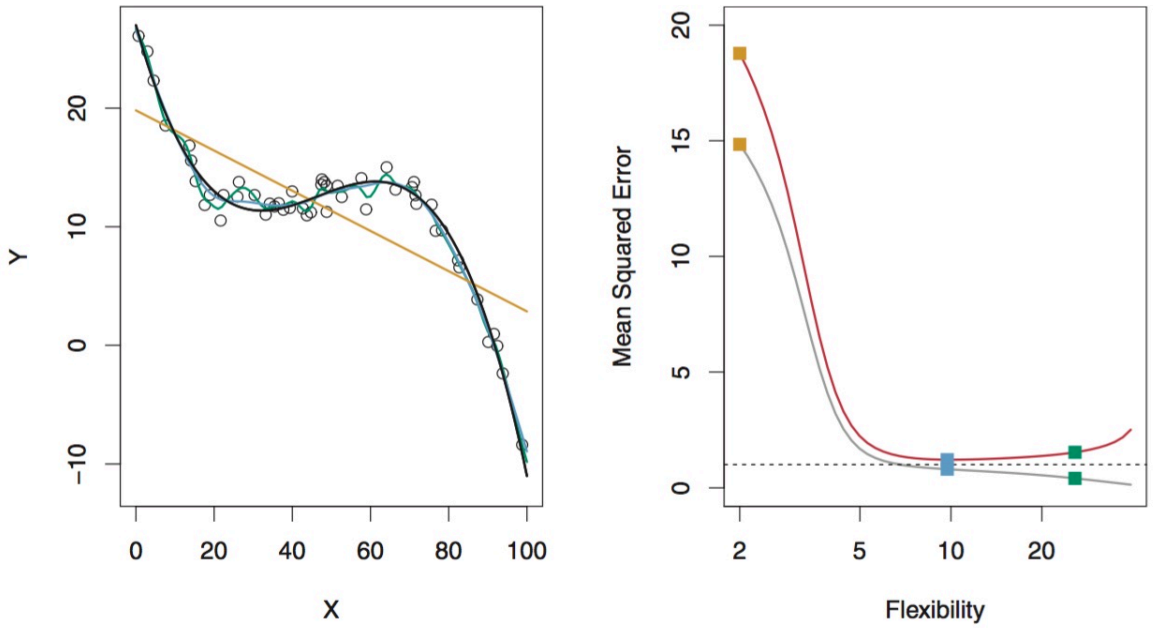


FIGURE 2.11. *Details are as in Figure 2.9, using a different f that is far from linear. In this setting, linear regression provides a very poor fit to the data.*

The final figure shows the bias-variance decomposition for each of these three models as a function of increasing complexity.

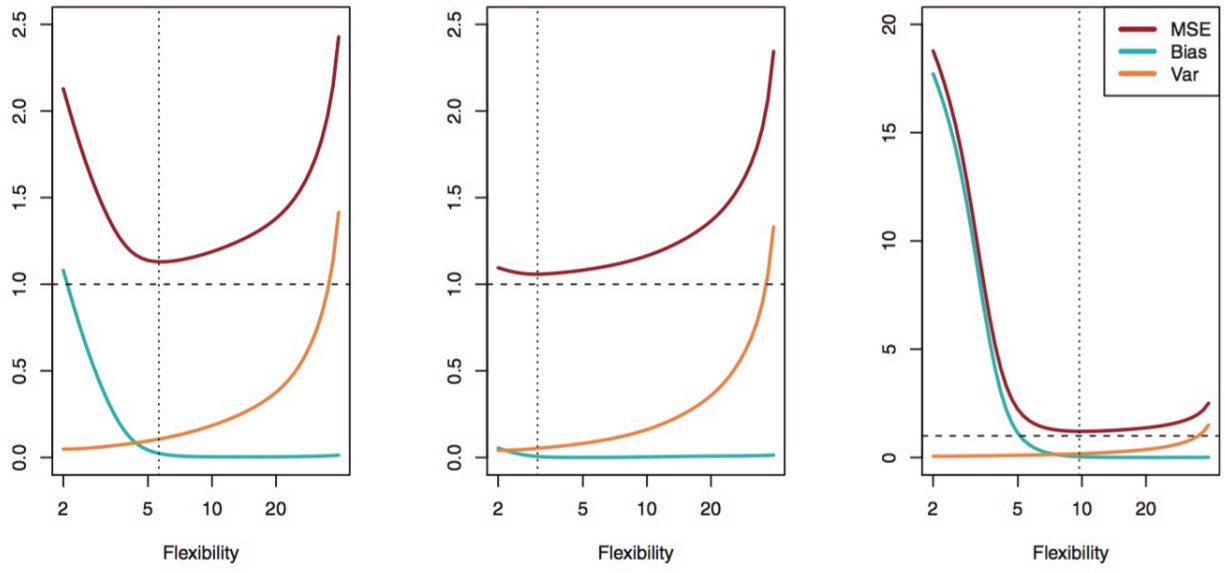


FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), $Var(\epsilon)$ (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

Additional Notes

You can find a very readable article about this topic by Scott Fortmann-Roe, here
scott.fortmann-roe.com/docs/BiasVariance.html
You can also find a nice article about the double descent phenomenon by Mikhail Belkin et al, here
https://www.pnas.org/content/116/32/15849.short