annotated version

Machine Learning Course - CS-433

Bias-Variance Decomposition

Oct 11, 2018

changes by Rüdiger Urbanke 2017

changes by Martin Jaggi 2018

Last updated: October 11, 2018

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

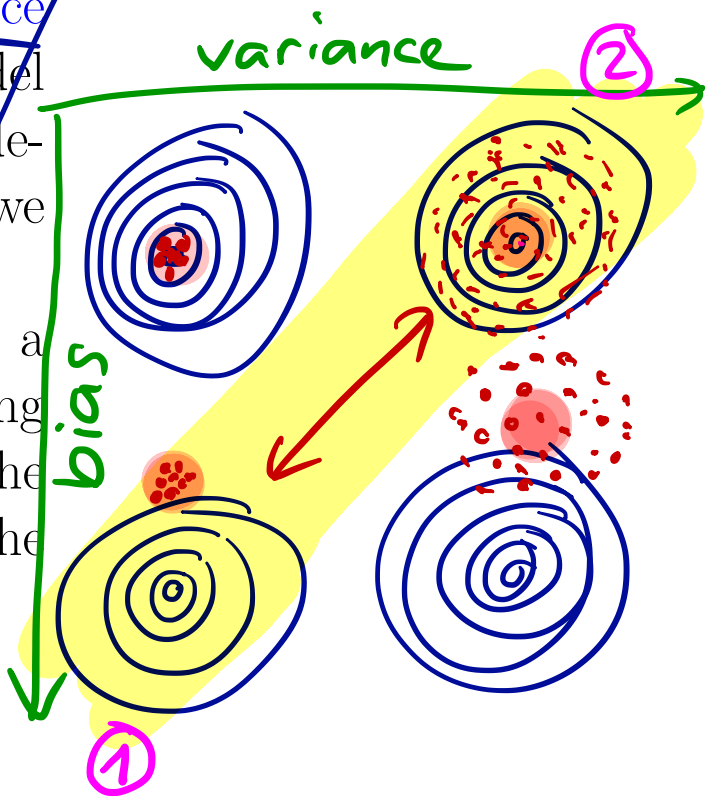
Motivation

Last time: how to choose good models and hyper-parameters. Split data into *train* and *test* parts.

Today: inherent bias - variance trade-off when we perform the model selection, i.e. when we need to decide how "complex" or "rich" we should make our model.

Example: linear regression with a one-dimensional input and using polynomial feature expansion. The maximum degree d regulates the complexity of the class.

same also for any other hyperparameter



1 Too

Simple model (e.g. small d):

- bad fit, underfit. large bias
- variance of $L_{\mathcal{D}}(f_S)$ (variations due to the random sample S) is small.
- large bias but a small variance

2 Too

Complex model (e.g. high d):

- good fit, overfitting. small bias
- high variance of $L_{\mathcal{D}}(f_S)$ as a function of S
- small 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.

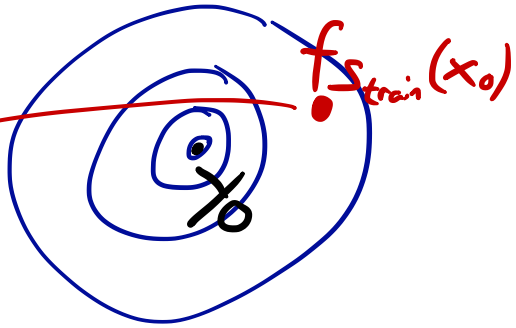
We further assume that $\mathbf{x}$ is generated according to some fixed but unknown distribution $\mathcal{D}_\mathbf{x}$. Finally, we assume that the loss function $\ell(\cdot, \cdot)$ is the square loss. Let $\mathcal{D}$ denote the joint distribution on pairs $(\mathbf{x}, y)$.

Fix test point (x_0, y_0)

Error Decomposition

$S_{train} = \{(x_n, y_n)_{n=1}^{N}\}$

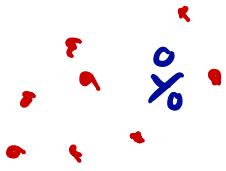
As always, we have given some training data S_{train}, consisting of iid samples according to $\mathcal{D}$. Given our learning algorithm $\mathcal{A}$, we compute the prediction function $f_{S_{\text{train}}} = \mathcal{A}(S_{\text{train}})$. We look at the square loss of this prediction function for a fixed element $\mathbf{x}_0$, i.e., we compute



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

where we used our specific data generation model.

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:

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

noise

randomly select training set of size N

$\mathrm{Var}(\varepsilon) = \mathbb{E}\,(\varepsilon - \underbrace{\mathbb{E}[\varepsilon]}_{=0})^2$ for our noise

$\mathbb{E}(u+v)^2 = \mathbb{E}u^2 + \mathbb{E}2u\cdot v + \mathbb{E}v^2$

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

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

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

$$\overset{(b)}{=} \mathrm{Var}_{\varepsilon\sim\mathcal{D}_\varepsilon}[\varepsilon] + \mathbb{E}_{S_{\text{train}}\sim\mathcal{D}}\big[\big(f(\mathbf{x}_0) - f_{S_{\text{train}}}(\mathbf{x}_0)\big)^2\big]$$

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

$+\mathbb{E}\,2u\cdot v = \mathbb{E}_{S,\varepsilon}\,\varepsilon\cdot(f - f_S) = (\mathbb{E}_\varepsilon \varepsilon)\cdot(\mathbb{E}f - f\ldots) = 0$

true — variance — bias — y_0

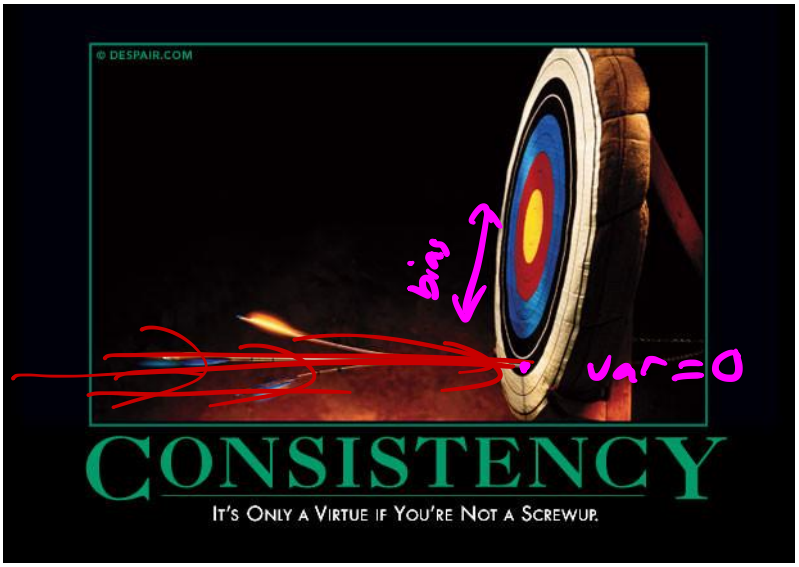
Note that here S'_{train} is a second training set, also sampled from $\mathcal{D}$ that is independent of the training set S_{train}.

$\mathbb{E}_{S\sim\mathcal{D}}\Big[\big(A - \mathbb{E}_{S'}[f_{S'}(x_0)] + \mathbb{E}_{S'}[f_{S'}(x_0)] - B\big)^2\Big]$

u + v

u^2 = bias $\qquad$ v^2 = variance

$+ 2\cdot u\cdot v$ ..... see next



Details:

In step (a), we omitted the third term

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

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 substracted 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 'train'-subscript)

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

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.

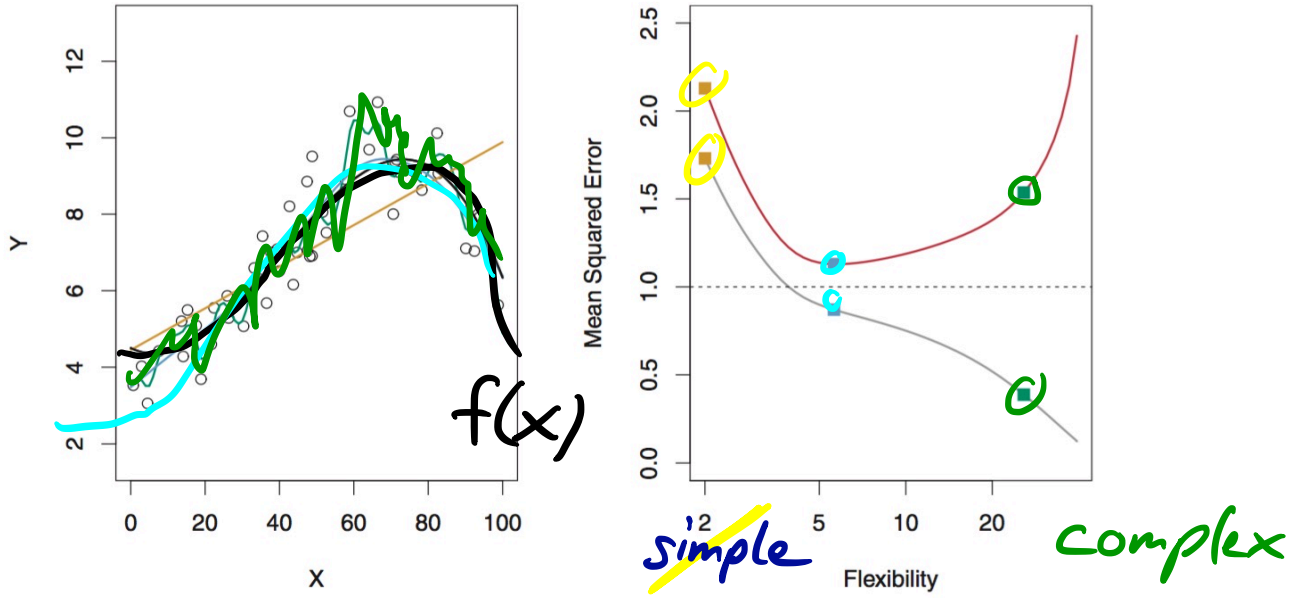


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed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

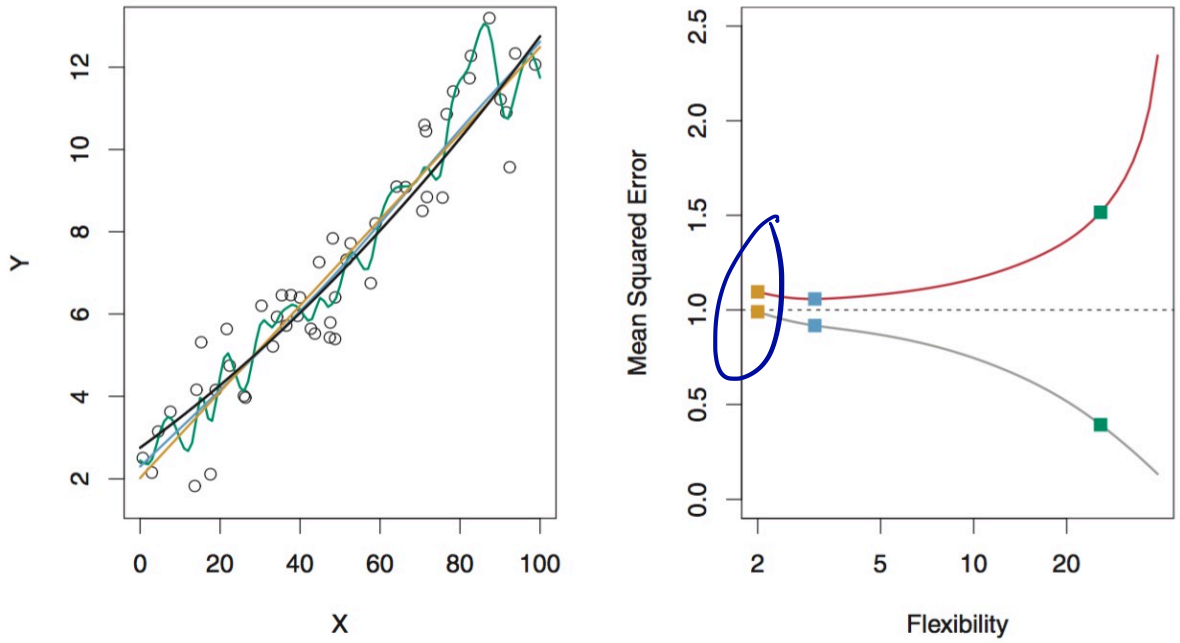


FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

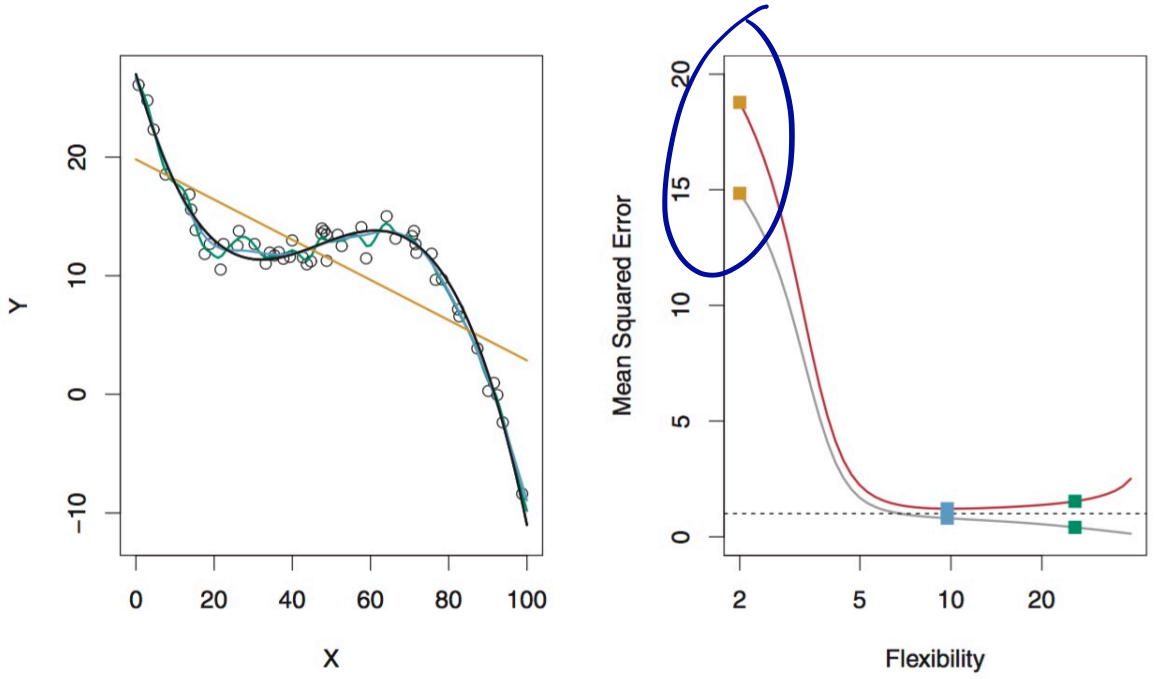


FIGURE 2.11. *Details are as in Figure 2.9, using a different f that is far from linear. In this setting, linear regression provides a very poor fit to the data.*

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

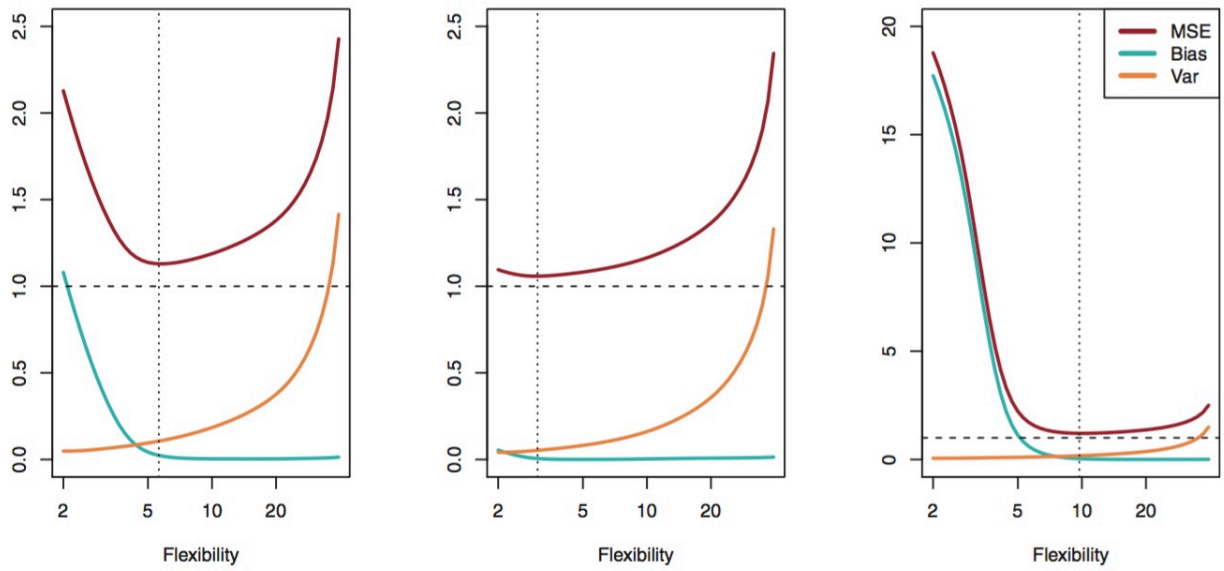


FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), Var(ϵ) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

Additional Notes

You can find a very readable article about this topic by Scott Fortmann-Roe, here
scott.fortmann-roe.com/docs/BiasVariance.html