Machine Learning Course - CS-433

Bias-Variance Decomposition

Oct 12, 2017

major changes by Rüdiger Urbanke 2016

changes by Rüdiger Urbanke 2017

Last updated: October 24, 2017

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Motivation

In the last lecture we have discussed how we can choose good hyper-parameters. The basic idea was to split the data and to use one part to *train* and the other to *validate*.

Today we will talk about an inherent bias - variance trade-off that we are facing when we perform the model selection. Recall that the underlying problem is to decide how "complex" or "rich" we should make our model.

To be concrete, consider a regression problem with a one-dimensional input and a polynomial basis as features. In this case the maximum degree we allow, call it d, regulates the complexity of the class.

But the same principle applies whenever we have a parameter that regulates the complexity of the model (e.g., in the ridge regression problem this is the parameter λ).

As we have discussed, when d is large (think $d \to \infty$) we are considering a very rich model. But if d is small (think $d = 0, 1$) then the model is very constrained. In both these extreme cases we are likely going to see poor performance, but for very different reasons.

If d is small then it is likely that we cannot find a good prediction function within our model to fit the data. We are saying that we are having a large *bias* (bad fit, underfit). But since we are only considering very simple models (e.g., a constant, or affine function) even with relatively little data S, we are likely to *always* learn essentially the same function f_S and so the associated loss will vary little as a function of S. Hence the variance of $L_{\mathcal{D}}(f_S)$ (the variations due to the random sample S) is small. In summary: for simple models we expect a large bias but a small variance.

Consider now the other extreme where d is very large (very complex model). In this case there are likely functions contained in the model that result in a very good fit with the given data and so we expect a small bias. But we are in danger of overfitting the data. Even changing a single sample from S_t might change the resulting prediction function f_S considerably. We therefore expect to see a high variance of $L_{\mathcal{D}}(f_S)$ as a function of S. In summary: for complex models we expect small bias but high variance.

Data Generation Model

Let us now make this intuition precise. Assume that the data is generated in the following way. Let

$$y = f(\mathbf{x}) + \epsilon, \tag{1}$$

where f is some (arbitrary and unknown) function and ϵ is additive *noise* with distribution $\mathcal{D}_\epsilon$ that is independent from sample to sample and independent from the data. Without loss of generality we can assume that 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_t, consisting of iid samples according to $\mathcal{D}$. Given our learning algorithm $\mathcal{A}$, we compute the prediction function $f_{S_t} = \mathcal{A}(S_t)$. We look at the square loss of this prediction function for a fixed element $\mathbf{x}_0$, i.e., we compute

$$(f(\mathbf{x}_0) + \epsilon - f_{S_t}(\mathbf{x}_0))^2,$$

where we used our specific data generation model.
We imagine that we are running the experiment many times: we create S_t, we learn the model f_{S_t}, 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:

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

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

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

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

In step (a), besides the two terms that we have written down we also have the term

$$\mathbb{E}_{S_t \sim \mathcal{D}, \epsilon \sim \mathcal{D}_\epsilon}[2\epsilon(f(\mathbf{x}_0) - f_{S_t}(\mathbf{x}_0))].$$

But since the noise ϵ is independent from S_t 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 $\mathbb{E}_{S_t' \sim \mathcal{D}}[f_{S_t'}(\mathbf{x}_0)]$ 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)

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

Interpretation of Decomposition

Let us now interpret this decomposition. We first note that each of the three terms is non-negative. Hence each of them is a lower bound on the expected loss when we predict the value for the input $\mathbf{x}_0$.

In particular, if the data contains noise, as we have assumed, then this noise imposes a strict lower bound on what error we can achieve. This contribution is given by the term $\text{Var}_{\epsilon\sim\mathcal{D}_\epsilon}[\epsilon]$.

Next, consider the "bias term." It is the square of the difference between the actual value $f(\mathbf{x}_0)$ and the expected prediction $\mathbb{E}_{S'_t\sim\mathcal{D}}[f_{S'_t}(\mathbf{x}_0)]$, where the expectation is over the training sets.

As we discussed, if we only allow simple models then we will not be able to find a good fit for the function within our model class, and so will not find a good fit in average. Hence, the bias will be large. This bias adds to the error that we observe.

Finally we have the "variance" term. It 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 of the value associated to some input $\mathbf{x}_0$ will vary widely. This variance adds further to our total error.

Examples

The following four figures are take from the book by James, Witten, Hastie, and Tibshiranie (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.

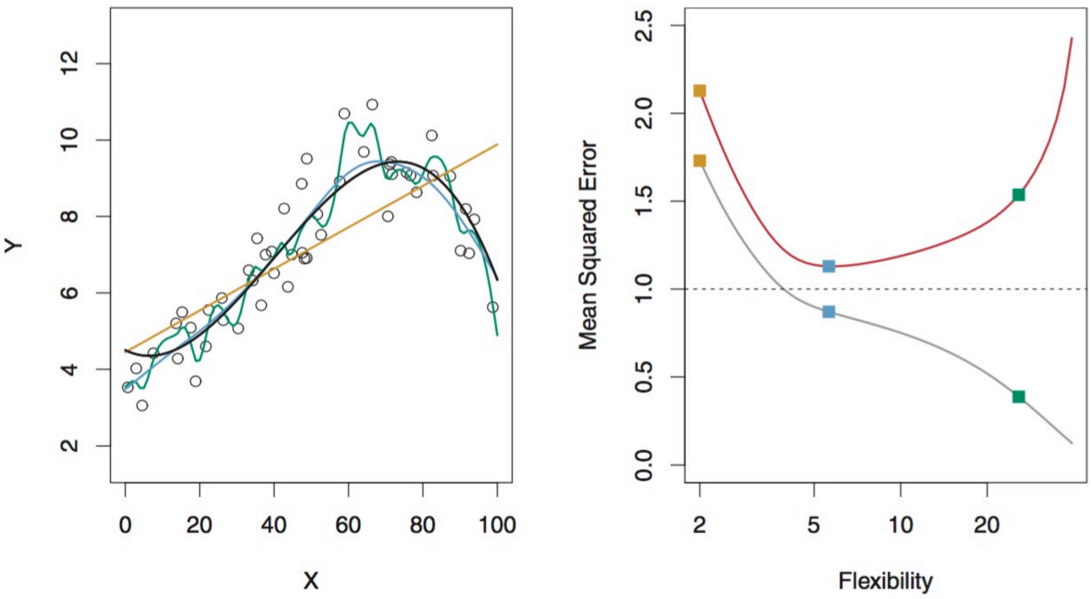

FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed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

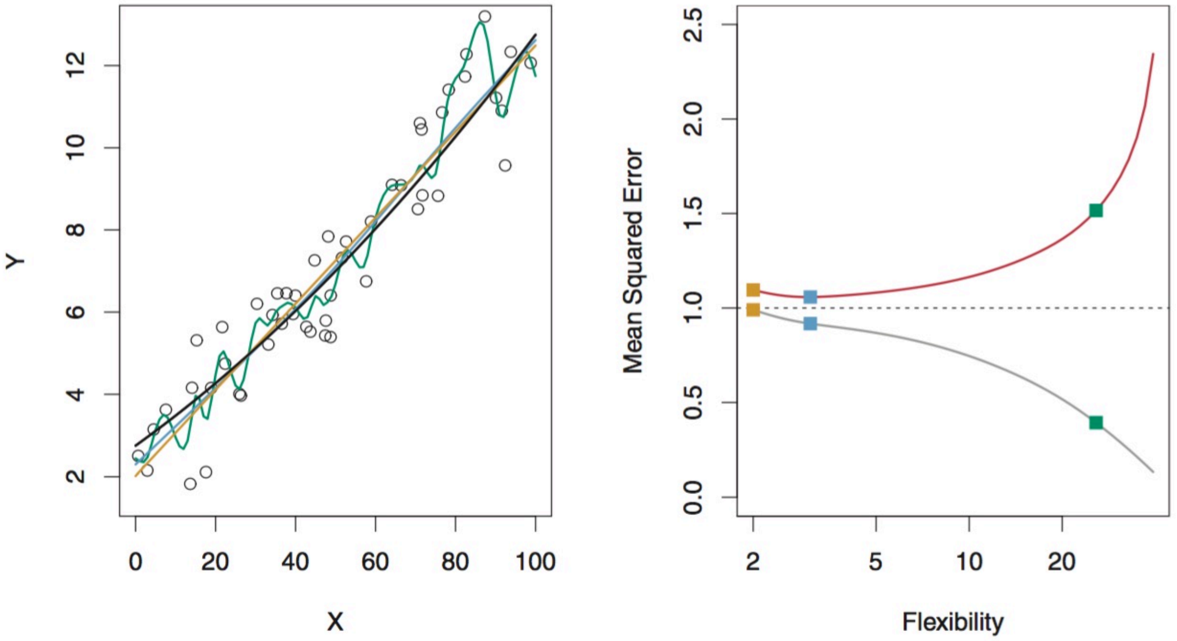

FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

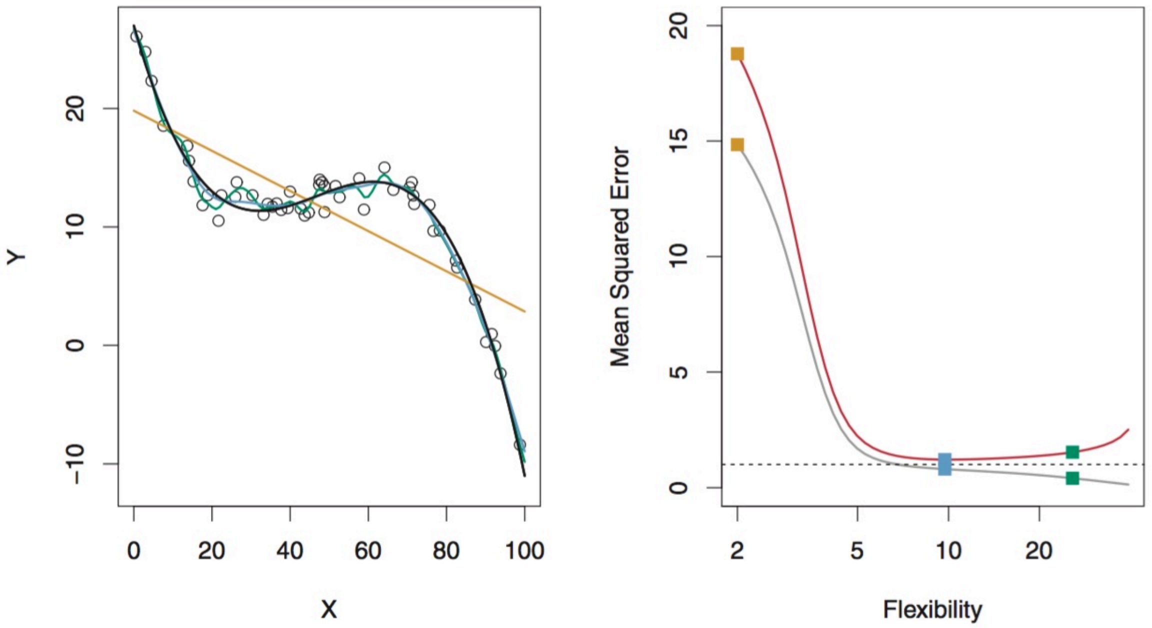

FIGURE 2.11. *Details are as in Figure 2.9, using a different f that is far from linear. In this setting, linear regression provides a very poor fit to the data.*

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

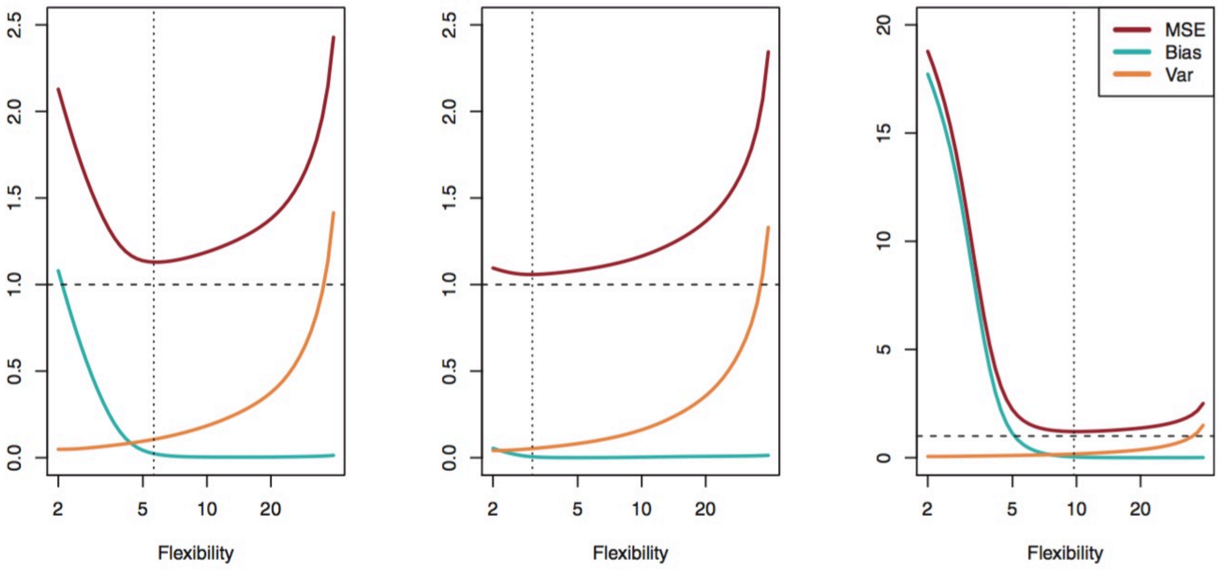

FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), Var(ϵ) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

Additional Notes

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