Lecture 2 - Models

Sondre Elstad

2022-06-25

Introduction

"All models are wrong - but some are useful!" This is ancient wisdom in almost any science. Naturally, we want our models to be as 'good' as possible. However, this implies a clear and unequivocal definition of what we mean by 'good'. In these notes, we will consider statistical models. By statistical models we shall mean "a mathematical model that embodies a set of statistical assumptions concerning the generation of sample data". Put differently, we would like for there to be a connection between the true data-generating process (i.e., the true mechanism or function that has generated the data we are observing) and the model we build using our assumptions. We cannot hope to capture all relevant aspects of the world in one single model. In fact, we will see that any attempt to do so would result in a grossly errant model. The reason is that no model we build can be better than the data we apply the model on. As we will see, increasing the complexity of a model necessarily implies a trade-off between bias and variance. As such, the natural starting point is to consider the purpose for which we are building a model in the first place. Models with high predictive power are generally different from models with high explanatory power. This will be our starting point for the following discussion.

Statistical Models

Suppose we observe a quantitative response Y and p different predictors, $X_1, X_2, \ldots, X_p$. We assume there is a relationship between the quantitative response and the predictors:

$$Y = f(X) + \epsilon.$$

Here, f is a fixed (yet unknown) function of $X = (X_1, ..., X_p)$, and ϵ is a random error term. We assume that ϵ is independent of X and has mean zero: $E(\epsilon) = 0$. We need this assumption, because in its absence there is no way for $f(X)$ to represent the true relationship between Y and X. Machine learning has to do with identifying and applying 'suitable' models for estimating $f(X)$. Inference has to do with estimating $f(X)$ in cases in which the outcome, Y, is observed.

Prediction, on the other hand, has to do with estimating $f(X)$ in the cases in which we do not observe Y. On a philosophical note, this is why we often talk about inferring the past and predicting the future. In cases where the outcome has been realized, we are inferring the causal mechanism that has yielded the observed outcome. Future outcomes have not yet been realized, however, and so we must take into consideration prediction error, i.e. differences between a predicted outcome $\hat{Y}$ and the realized outcome, Y. Since the error term has average (or mean) equal to zero, we try predicting Y using

$$\hat{Y} = \hat{f}(X),$$

where $\hat{f}$ is our estimate for f. In cases where the outcome is observed, we are generally very interested in the exact relationship between Y and $X = (X_1, X_2, \ldots, X_p)$. This is the way we are used to thinking about statistical models in science: as tools for inquiring into causal mechanisms between a set of regressors and some observed outcomes.

However, prediction problems are slightly different. In these cases, we do not observe Y. We want to make as good predictions as possible, i.e. we want to minimize the forecasting error. In some cases, we might be lucky: we have scientific theories which have been excessively tested and verified for many years and in with great generality. In these cases, our predictions exhibit very low prediction errors. Alas, more often than not, this is not the case in sciences which rely more on observational data than on experimental data (e.g., the social sciences). When making predictions about societal phenomenon - such as the outcome of the next election - we cannot rely excessively on a generalizable theory. We simply do not have the luxury of being very explicit about the assumed functional form of $f(X)$. We must treat $\hat{f}$ as a black box. Anything goes as long as expected forecast error is low.

The accuracy of $\hat{Y}$ depends on two quantities - reducible error and irreducible error. By opting for a more accurate statistical model, we are able to reduce the reducible error. This justifies spending time and resources on finding a good fit, i.e. a model $\hat{f}$ which closely resembles the true f. However, even if our model $\hat{f}$ is 100% correct, there might still be some forecasting error! We assume $E(\epsilon) = 0$, but this doesn't mean that $\epsilon = 0$ will always hold. The cases where ϵ deviates from its mean, this gives rise to the so-called irreducible error. No matter how good our model is, we exercise no power over ϵ. This becomes more visible if we have a look at the squared expected forecasting error:

$$E(Y - \hat{Y})^2 = E[f(X) + \epsilon - \hat{f}(X)]^2 = \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{Var(\epsilon)}_{\text{Irreducible}},$$

If $Var(\epsilon) = 0, E(\epsilon) = 0$ would always be satisfied. In this case, there wouldn't be any irreducible error. However, this is generally not true (less so for observational data than for experimental data).

Interpretability vs. complexity

The figure below illustrates a trade-off. We see that there is a clear, negative correlation between model flexibility and model interpretability. In causal inference, we are generally very set on the importance of interpretable models. We want models with clear causal mechanisms. Linear models (found in the upper left corner under the headings 'Subset Selection', 'Lasso' and 'Least Squares') are easy to interpret because there is a clear linear relationship between the features and the outcome.

On the other hand, more flexible models like 'Support Vector Machines' and 'Deep Learning' are very flexible. They are very hard to interpret (which would explain the lack of interest for such models in causal inference). Flexible models are certainly very useful in many circumstances. Deep Learning allows us to solve tasks which we previously wouldn't be able to. However, if our intent is to make good predictions we would usually be better served going for a less flexible model than those found in the lower right corner. Why, you might ask?

There is no such thing as a free lunch! This is true for statistics as well. There doesn't exist any model which always outperforms all others (if so, we would always use this model, wouldn't we?) If we are looking for the model which provides the best predictions for a given data set, we need some metric or other for assessing this. Mean squared error (MSE) is such a metric:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

where $\hat{f}(x_i)$ is the prediction that $\hat{f}$ gives for the ith observation. When training a machine learning algorithm, we split our data set into two smaller data sets: the training set and the test set. We can compute MSE for the training set (training MSE) and for the test set (test MSE). We train our estimate $\hat{f}$ using the training set $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$. Based on this $\hat{f}$, we compute $\hat{f}(x_1), \hat{f}(x_2), ..., \hat{f}(x_n)$.

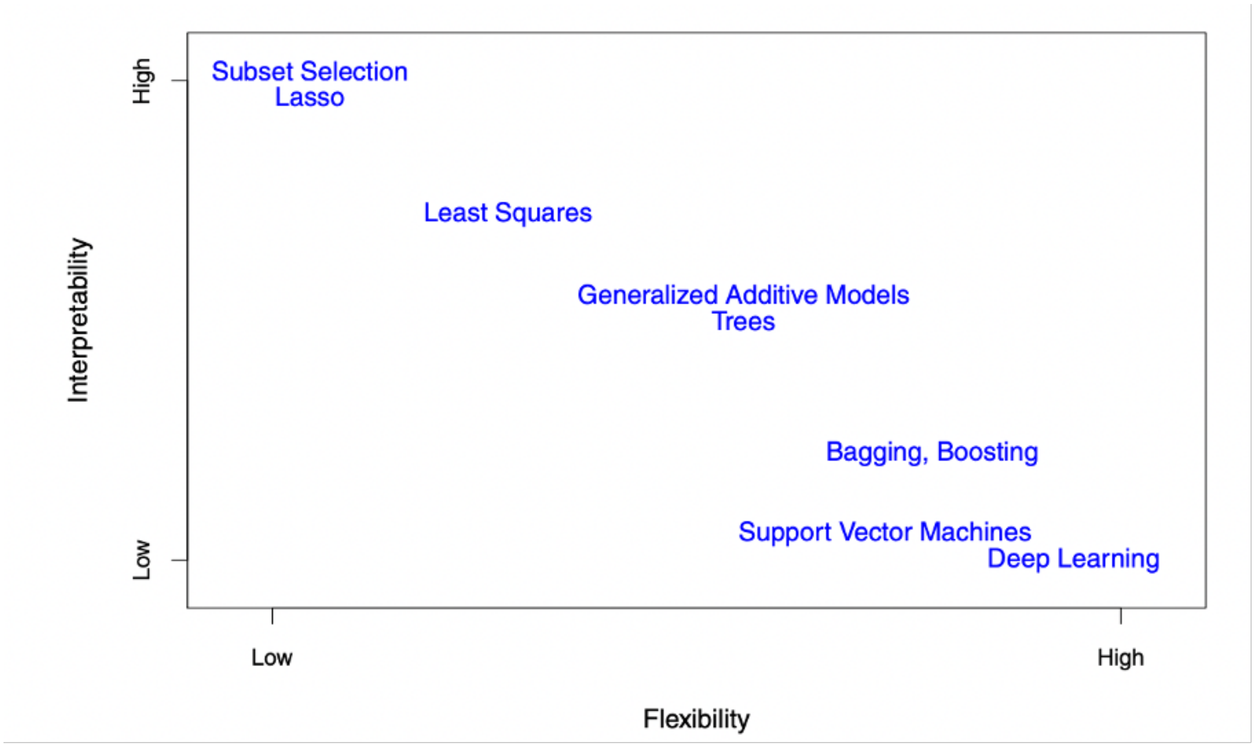

Figure 1: Trade-off between flexibility and interpretability. Generally, as flexibility of a method increases, its interpretability decreases. Source: ISL (2017).

If these are close to $y_1, y_2, ..., y_n$, the training MSE will be small. However, the proof lies in the pudding. In our case, the pudding is how well our $\hat{f}$ performs on a previously unseen test observation (x_0, y_0). We want to choose a $\hat{f}$ which yields a small test MSE. This has consequences for the optimal complexity (or flexibility) of our model.

The second chart shows that training MSE is monotonically decreasing in flexibility, whereas test MSE is exhibiting a U-shape in flexibility. We refer to the case with a small training MSE and a high test MSE as 'overfitting'. Overfitting occurs when we train our learning algorithm 'too' much. Flexible methods have the ability to draw a complex network between training observations. This implies a very low bias. However, there is no guarantee that the test data resembles the training data. We need a learning algorithm which takes into consideration that new observations might be different from previously seen observations. Put differently, we need a learning algorithm which puts some emphasis on the variance of the data-generating process. One might also think of it as requiring a statistical model which recognizes that training the model (or learning algorithm) on two different training sets generally yields somewhat different estimates. As such, we cannot minimize bias over the training set. We must balance bias and variance. This becomes clear as we can decompose expected test MSE into three different components:

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon).$$

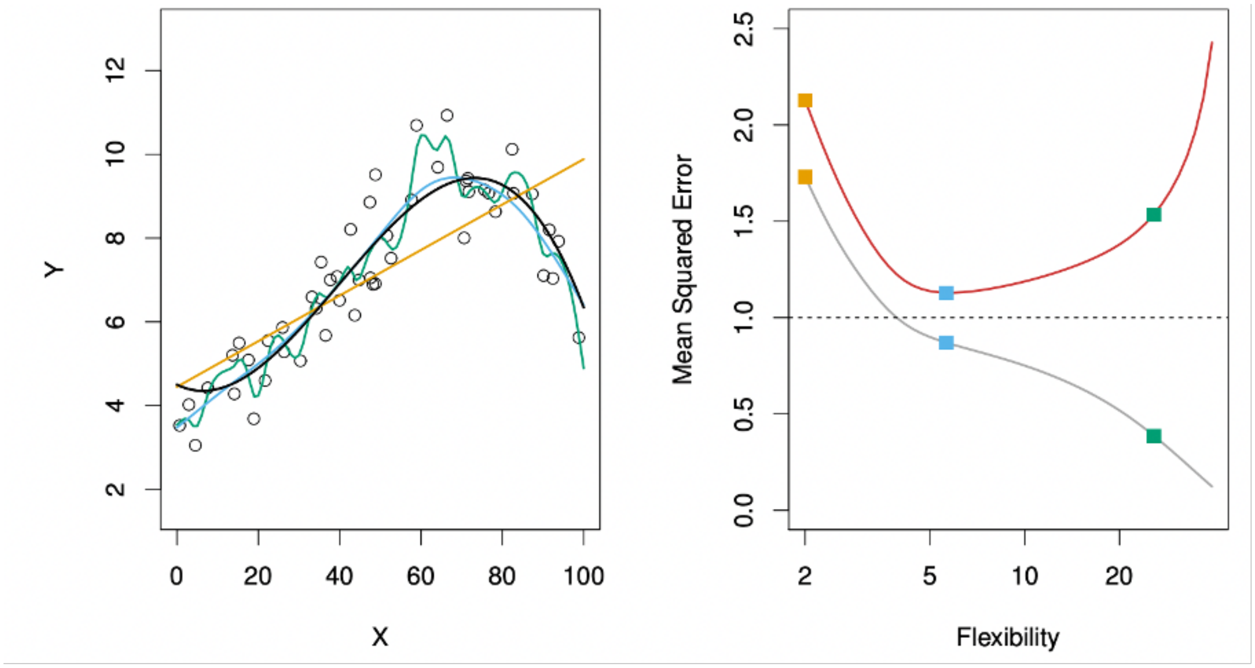

Figure 2: Left: Data simulated from some true f are shown in black. Three estimated f are shown: 1) linear regression (orange curve), 2-3) two smoothing splines (blue and green curves). Right: Training MSE (grey curve) and test MSE (red curve). Training MSE is monotonically decreasing in flexibility, whereas test MSE is concave in flexibility. Source: ISL (2017).