NBA 4920/6921 Lecture 7

Prediction Errors & the Variance-Bias Trade-Off

Murat Unal

Johnson Graduate School of Management

9/21/21

Agenda

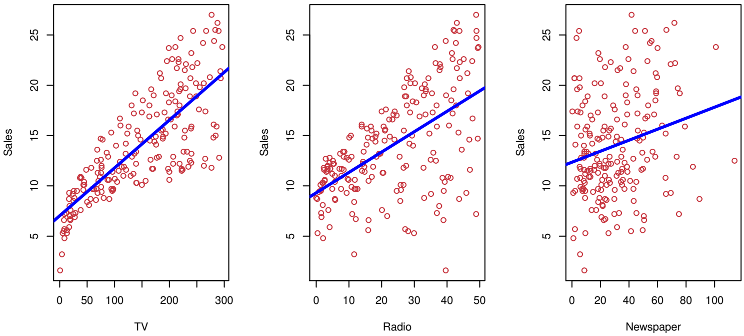
Quiz 6

Statistical learning

- Prediction error and loss
- Training vs testing
- Model fit

Variance-bias trade-off

Statistical learning



Source: ISL

Goal: Build a model to understand `Sales` as a function of advertisement spent.

Output/Response/Dependent Variable: $Y = $ `Sales`

Input/Feature/Predictor/Explanatory Variable: $X = ($`TV`,`Radio`,`Newspaper`$)$

Statistical learning

The relationship between output Y and p inputs, $X = (X_1, \cdots, X_p)$, can be written as

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

- f is an unknown function we want to learn/estimate
- It represents the **systematic** information that X provides about Y
- ϵ is a mean-zero error term that is independent of the inputs
- It represents the **noise/randomness/unobservables** that can not be explained using X

What can we use $\hat{f}$ for?

$$\texttt{Sales} = \hat{f}(\texttt{TV}, \texttt{Radio}, \texttt{Newspaper})$$

Using the observed data we learn/estimate f and obtain $\hat{f}$ for two main purposes:

1. **Inference**: Is higher advertising expenditure associated with higher sales? Which media contributes more?
2. **Prediction**: Predict sales from advertising expenditure.

Prediction error and loss

For regression problems, **Prediction error** is the difference between Y and its prediction $\hat{Y}$.

Loss is the distance (i.e., non-negative value) between a true value and its prediction.

$$\textbf{error}_i = y_i - \hat{y}_i \qquad\qquad \textbf{loss}_i = |y_i - \hat{y}_i|$$

Prediction error and loss

Loss functions aggregate and quantify loss.

L1: $\sum_i |y_i - \hat{y}_i|$

L2: $\sum_i (y_i - \hat{y}_i)^2$

Mean abs. error: $\frac{1}{n}\sum_i |y_i - \hat{y}_i|$

Mean squared error: $\frac{1}{n}\sum_i (y_i - \hat{y}_i)^2$

For classification problems, we use the **error rate** $\frac{1}{n}\sum_i \mathbb{1}(y_i \neq \hat{y}_i)$

Prediction error and loss

Both loss functions assume the following:

1. Overestimating is equally bad as underestimating
2. Errors are similarly bad for all observations

Prediction error and loss

They only differ in their assumptions about the magnitude of errors:

- **L1:** an additional unit of error is equally bad everywhere
- **L2:** an additional unit of error is worse when the error is already big

Prediction error and loss

The accuracy of $\hat{Y}$ as a prediction for Y depends on two quantities:

1. **Reducible error:** The error that we can reduce and improve the accuracy of $\hat{f}$
2. **Irreducible error:** The error that is introduced by ϵ, we do not measure/observe it, hence we can not reduce it.

Prediction error and loss

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

We can never have $Y = \hat{Y}$, even if we know f

All the techniques we will discuss aim to minimize the **reducible error** for learning/estimating $\hat{f}$

Model performance

A linear model is restrictive, can have lower prediction accuracy, but more interpretability

$\rightsquigarrow$ easier understanding the relationship between Y and X

Non-parametric and non-linear models are highly flexible, can lead to more accurate predictions, but are harder to interpret

Model performance

How do we choose between competing models?

We need a measure to define model performance

In regression settings we use the Mean Squared Error (MSE):

$$\frac{1}{n}\sum_{i=1}^{n}[\underbrace{y_i - \hat{f}(x_i)}_{\text{prediction error}}]^2$$

For classification problems, we use the Classification Error Rate

$$\frac{1}{n}\sum_{i}\mathbb{1}(y_i \neq \hat{f}(x_i))$$

Training vs testing

MSE is computed using the training data we used to fit the model.

We want it to be low.

But we are more interested in prediction accuracy for data that we have not seen before, the test data.

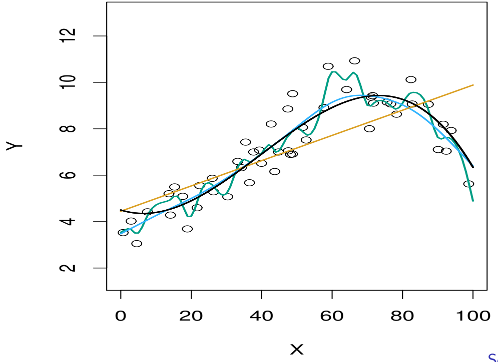
Training vs testing

If you want to build a model to predict stock market performance, you train a model with historic stock market data, but you want to predict the next day's performance.

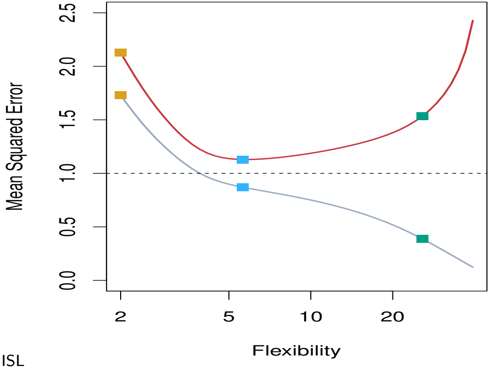
We want to choose the method that gives the lowest MSE in the test data.

In other words we aim for high generalizability or external validity.

Model fit



Source: ISL



Black: True $f(x)$

Orange: Linear regression fitted

Blue and green: Splines fitted

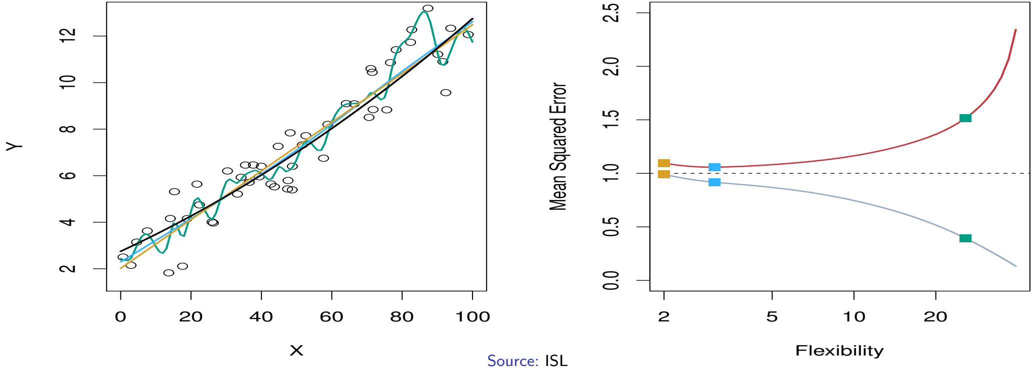
Red: Test MSE curve

Grey: Training MSE curve

Dashed: $Var(\epsilon)$ = Min. test MSE

Green spline is overfitting, we can achieve lower test MSE with a less flexible model - blue spline

Model fit



Source: ISL

Black: True $f(x)$

Orange: Linear regression fitted

Blue and green: Splines fitted

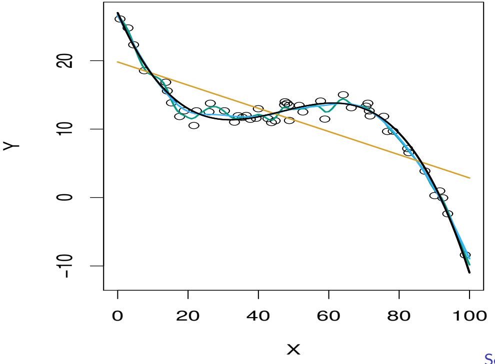
Red: Test MSE curve

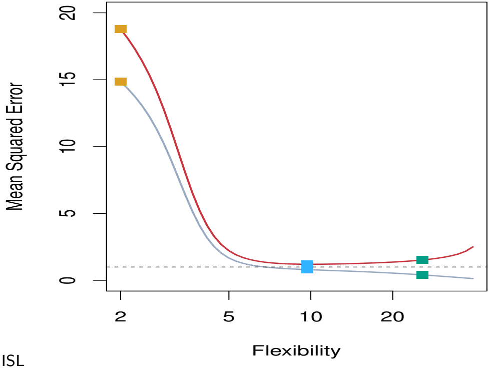
Grey: Training MSE curve

Dashed: $Var(\epsilon)$ = Min. test MSE

Because the truth is linear, linear regression fits well.

Model fit





Source: ISL

Black: True $f(x)$

Orange: Linear regression fitted

Blue and green: Splines fitted

Red: Test MSE curve

Grey: Training MSE curve

Dashed: $Var(\epsilon)$ = Min. test MSE

The truth is highly non-linear, linear regression fits very poor.

Variance-bias trade-off

The U-shape observed in the test MSE curves turns out to be the result of two competing properties of statistical learning methods.

The expected test MSE, for a given value x_0, consists of three quantities:

$$E(y_o - \hat{f}(x_0))^2 = \underbrace{Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

Variance-bias trade-off

$$E(y_o - \hat{f}(x_0))^2 = \underbrace{Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

We can obtain the expected test MSE using a large number of training sets and repeatedly estimating f and then test each at x_o

In order to minmize it we need a method that can achieve both **low variance** and **low bias** of $\hat{f}(x_0)$

Variance-bias trade-off

$$E(y_o - \hat{f}(x_0))^2 = \underbrace{Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

Variance refers to the amount by which $\hat{f}$ would change if we estimated it using a different data set

If a method has high variance then small changes in the data will result in large fluctuations in $\hat{f}$

More flexible methods have higher variance

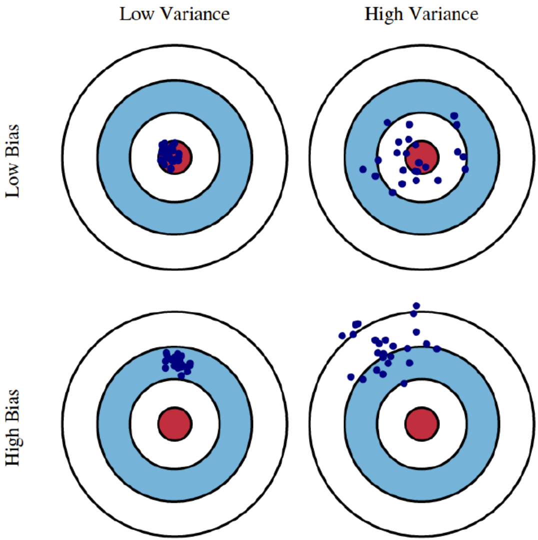
Variance-bias trade-off

$$E(y_o - \hat{f}(x_0))^2 = \underbrace{Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

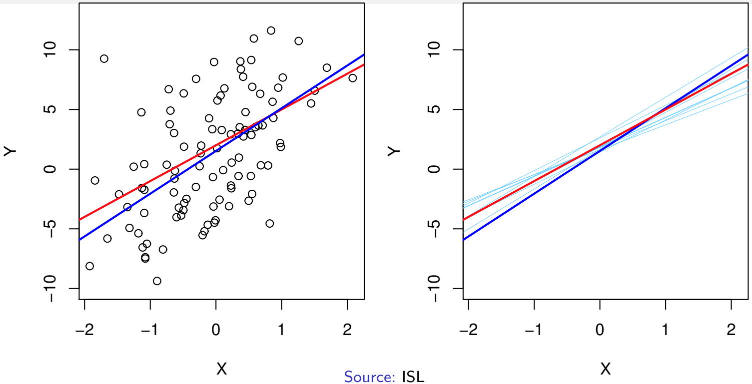
Bias refers to the error that is introduced from inaccurately estimating f

Simple methods will result in higher bias - real life is messy, most likely not linear

Variance-bias trade-off



Variance-bias trade-off



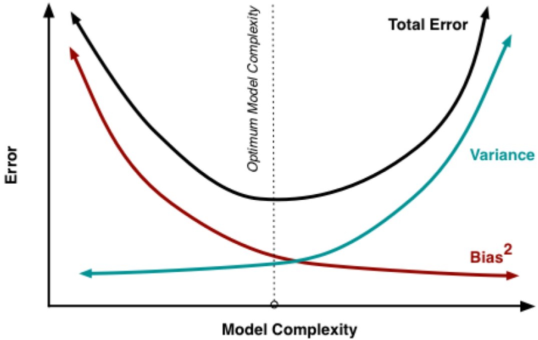
Source: ISL

Red line is the true relationship: $f(X) = 2 + 3X$

Other lines are least squares estimates for $f(X)$, each obtained by fitting a different sample

Each line is different, but in this case on average, the lines are close to the red line

Variance-bias trade-off



Source: ISL

Initially increasing flexibility reduces bias more than it increase variance, which leads to smaller test MSE

Optimal model flexibility is achieved when the marginal benefits of flexibility equal marginal costs

Variance-bias trade-off

$$E(y_o - \hat{f}(x_0))^2 = \underbrace{Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

The expected test MSE can never lie below $Var(\epsilon)$

Q: Why?

Variance-bias trade-off

$$E(y_o - \hat{f}(x_0))^2 = \underbrace{Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2}_{Reducible} + \underbrace{Var(\epsilon)}_{Irreducible}$$

The expected test MSE can never lie below $Var(\epsilon)$

Q: Why?

A: Because $Var(\hat{f}(x_0)) \geq 0$ and $[Bias(\hat{f}(x_0))]^2 \geq 0$

Recap

Statistical learning requires careful consideration of various tradeoffs:

- Model complexity and interpretability
- Performance in training and test data
- Variance and bias

Recap

Supervised learning:

1. We define a model for the relationship between the observed data, $Y = f(X) + \epsilon$, and train the model to obtain the estimate $\hat{f}$
2. We use the trained model to obtain MSE

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[\underbrace{y_i - \hat{f}(x_i)}_{\text{prediction error}}]^2$$

3. Our goal is to use the method that achieves low MSE on data that the model has not seen before, i.e. test data

Sources

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani (2017)
An Introduction to Statistical Learning
Springer.
https://www.statlearning.com/