estimand

An introduction to prediction

Dr Gianluca Campanella

19th May 2016

Contents

Prediction and loss functions

Exercise: guessing values

Consider the random variable Y representing the time it takes you to get here.

You have some data which you collected over the last three weeks, and want to estimate how long it will take you to get here on Tuesday.

How do you do this?

If you prefer, what is the **optimal point forecast** for X?

Loss functions

Before you can answer the previous question, you need to pick a **loss function** that...

- Measures how big an error you're making
- Can be minimised

Loss functions

Before you can answer the previous question, you need to pick a **loss function** that...

- Measures how big an error you're making
- Can be minimised

Squared loss
(mean squared error)

$$\text{MSE}(g) = \mathbb{E}\left[(Y - g)^2\right]$$

Absolute loss
(mean absolute error)

$$\text{MAE}(g) = \mathbb{E}[|Y - g|]$$

Towards prediction...

- Predicting a single number g is not very useful
- Usually we have at least another variable X that we believe to be related to Y *somehow*

Towards prediction...

- Predicting a single number g is not very useful
- Usually we have at least another variable X that we believe to be related to Y *somehow*

Idea

Using X, we should be able to predict **better** (i.e. reduce the mean error) than by ignoring it

$$g \rightarrow f(X) \quad \text{and thus} \quad \text{MSE}(f) = \mathbb{E}\left[(Y - f(X))^2\right]$$

Optimal prediction

What should f be if we want to know Y given X?

Consider the decomposition

$$Y \mid X = f^\star(X) + \varepsilon$$

- $f^\star$ is the optimal conditional prediction
- ε is a random variable (since $f^\star$ is not)
- $\mathbb{E}[\varepsilon] = 0$ without loss of generality

Optimal prediction

Using the MSE, it can be shown that $f^\star$ is the conditional (on $X = x$) expected value:

$$f^\star(x) = \mathbb{E}[Y \mid X = x]$$

$f^\star$ is what we'd like to know when we want to predict Y

...but can we?

Bias-variance trade-off

Bias-variance trade-off

Suppose that...

- The 'true' regression function is $f^\star$
- But we have to make do with f

Let's start by expanding...

$$\begin{aligned}(Y - f)^2 &= (Y - f^\star + f^\star - f)^2 \\ &= ((Y - f^\star) + (f^\star - f))^2 \\ &= (Y - f^\star)^2 + 2(Y - f^\star)(f^\star - f) + (f^\star - f)^2\end{aligned}$$

Bias-variance trade-off

Now take the expectation...

$$\mathbb{E}\left[(Y - f^\star)^2 + 2(Y - f^\star)(f^\star - f) + (f^\star - f)^2\right]$$

Since $Y - f^\star = \varepsilon$...

- $\mathbb{E}\left[(Y - f^\star)^2\right] = \mathbb{V}[\varepsilon]$
- $\mathbb{E}[Y - f^\star] = \mathbb{E}[\varepsilon] = 0$
- $\mathbb{E}\left[(f^\star - f)^2\right] = (f^\star - f)^2$ (non-random)

Bias-variance trade-off

$$\mathrm{MSE}(f) = \mathbb{V}[\varepsilon] + (f^\star - f)^2$$

Variance $\mathbb{V}[\varepsilon]$

- Doesn't depend on f, just on 'how hard' it is to predict Y at $X = x$
- It is the unpredictable, irreducible fluctuation around even the best prediction

Bias-variance trade-off

$$\mathrm{MSE}(f) = \mathbb{V}[\varepsilon] + (f^\star - f)^2$$

Bias $(f^\star - f)^2$

- Is the 'extra error' we get from not knowing $f^\star$
- Also the amount by which we are *systematically* off

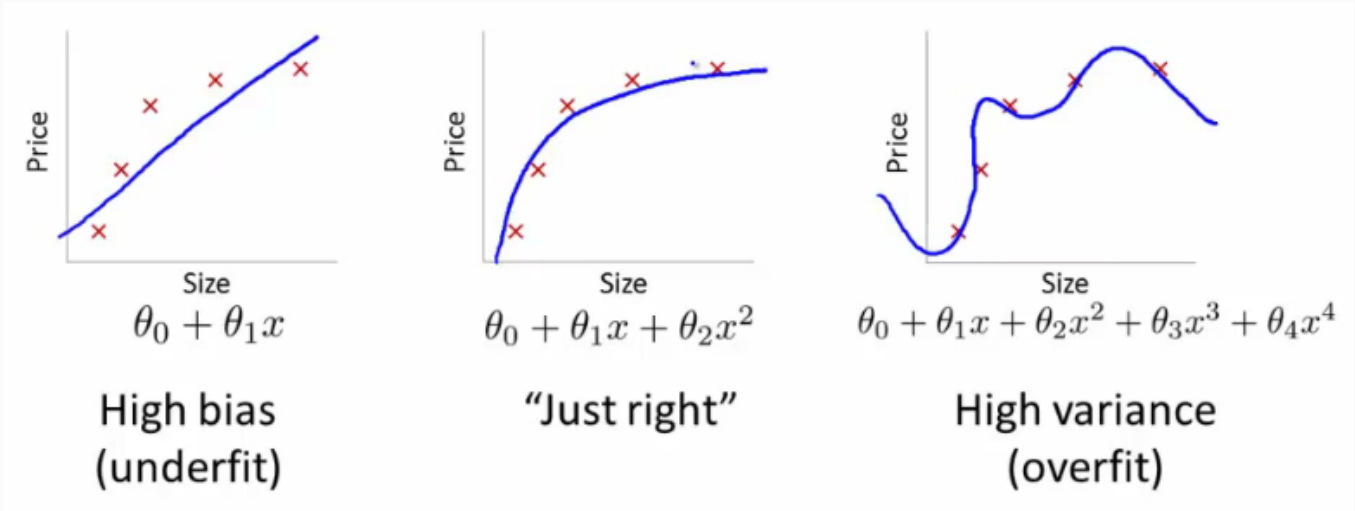
Bias-variance trade-off

Typically f is itself estimated from a sample, so we have...

- The **irreducible variance** due to the process
- The **bias** in approximating $f^\star$ using f
- The added **estimation variance** of $\hat{f}$

- A method is **consistent** when the bias and estimation variance go to zero as the sample size increases
- Different consistent methods may converge at different rates

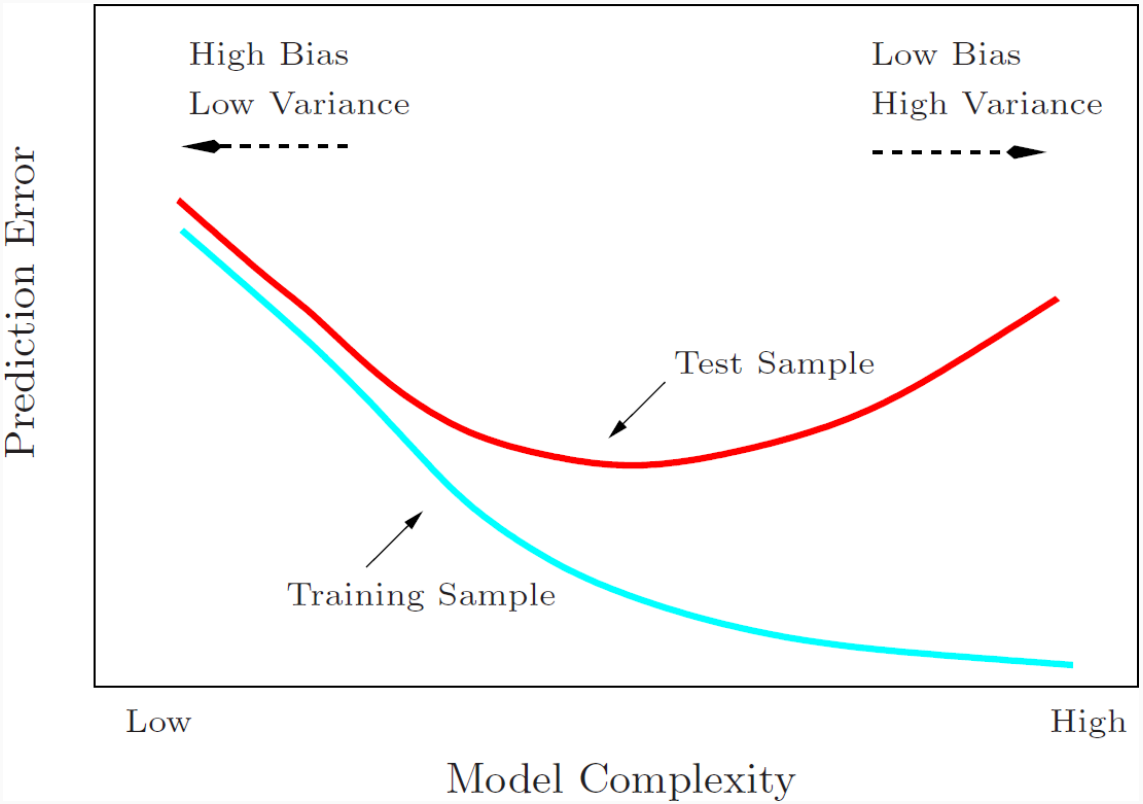
Bias-variance trade-off



(From Andrew Ng's ML course)

Generalisability

Bias-variance trade-off and generalisability



(From *The Elements of Statistical Learning*)

Cross-validation

General idea

- Fit several models on subsets of the data
- Measure performance of each
- Compute the mean performance

k-fold cross-validation

- Split the data into k groups (a.k.a. 'folds')
- Fit the model using all but one
- Test on the excluded subset

Regularisation

General idea

- Penalise ‘large’ coefficients by shrinking them
- Helps avoid overfitting
- Requires tuning of an additional parameter α representing the ‘weight’ of the penalty

L_1 penalty
(LASSO)

$$\sum_j |\beta_j|$$

L_2 penalty
(Tikhonov or ridge)

$$\sum_j \beta_j^2$$