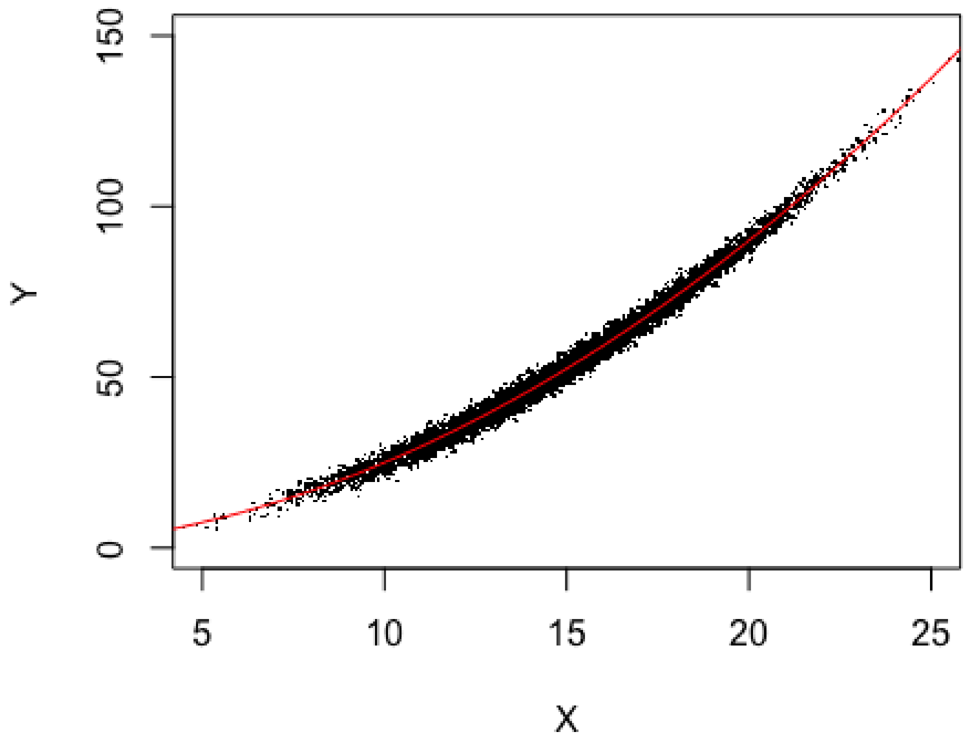
Consider the case where true relationship between Y and X is quadratic

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$$

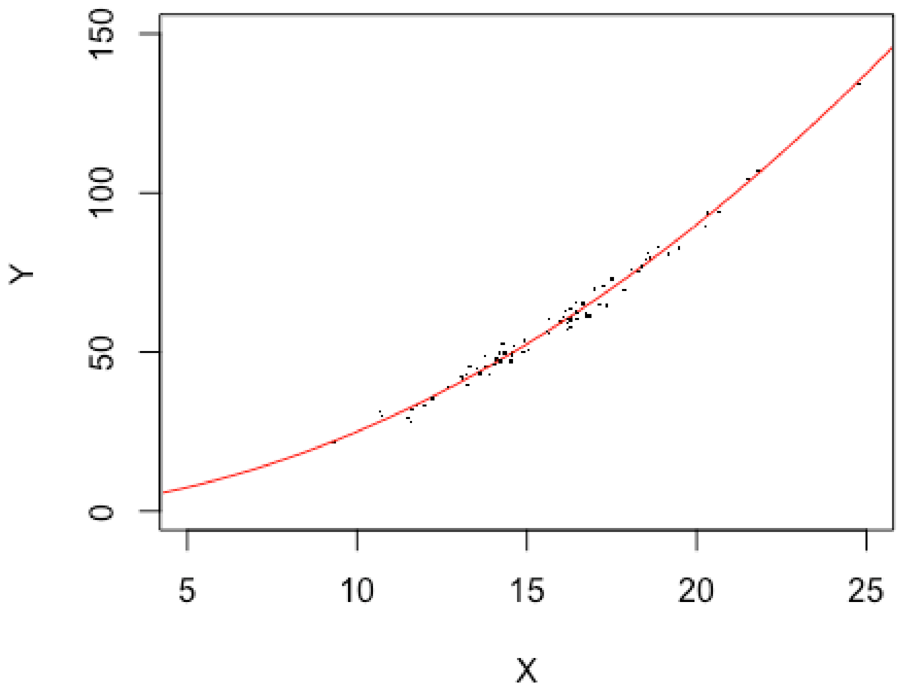


Plot shows full population,
true regression function

Consider the case where true relationship between Y and X is quadratic

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$$

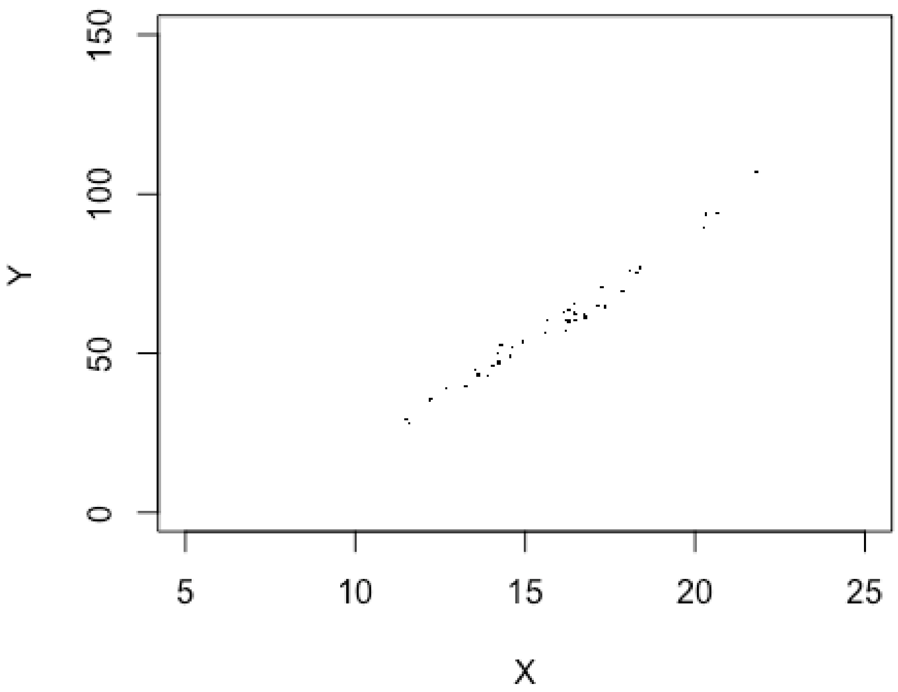
We only have a sample of the population to work with:



Plot shows 40 points,
true regression function

But we don't know the true regression function

Overfitting

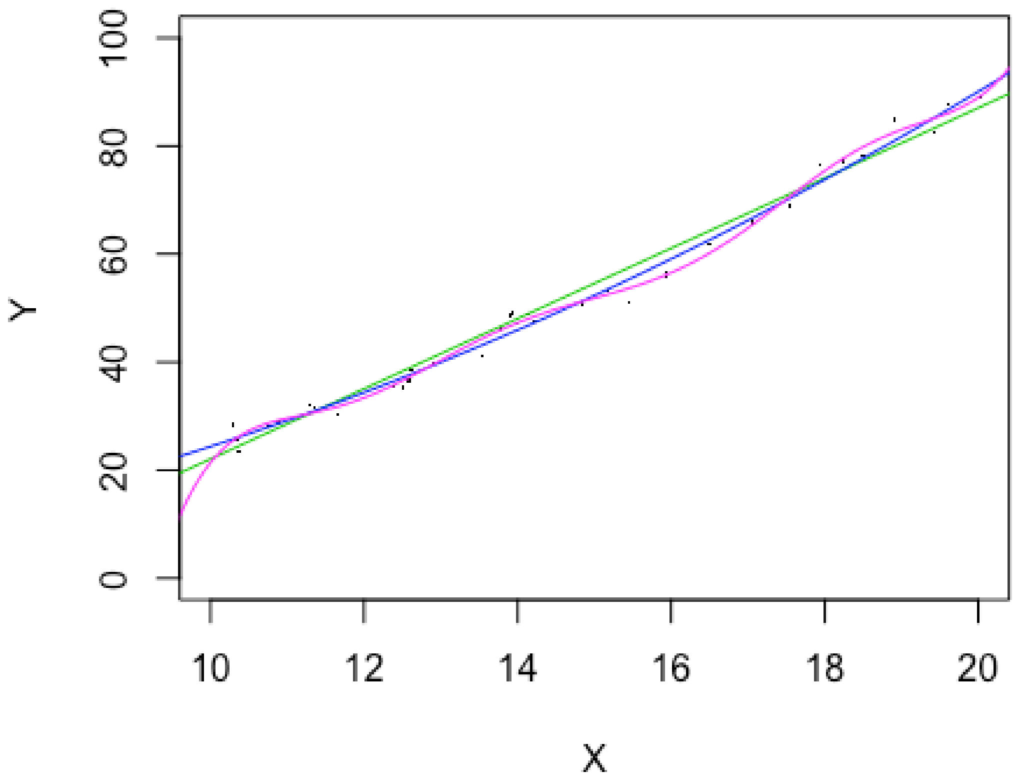


Plot shows 40 points,

To find regression function

try fits of different flexibility
(using different polynomials)

Overfitting



Plot shows 40 points,

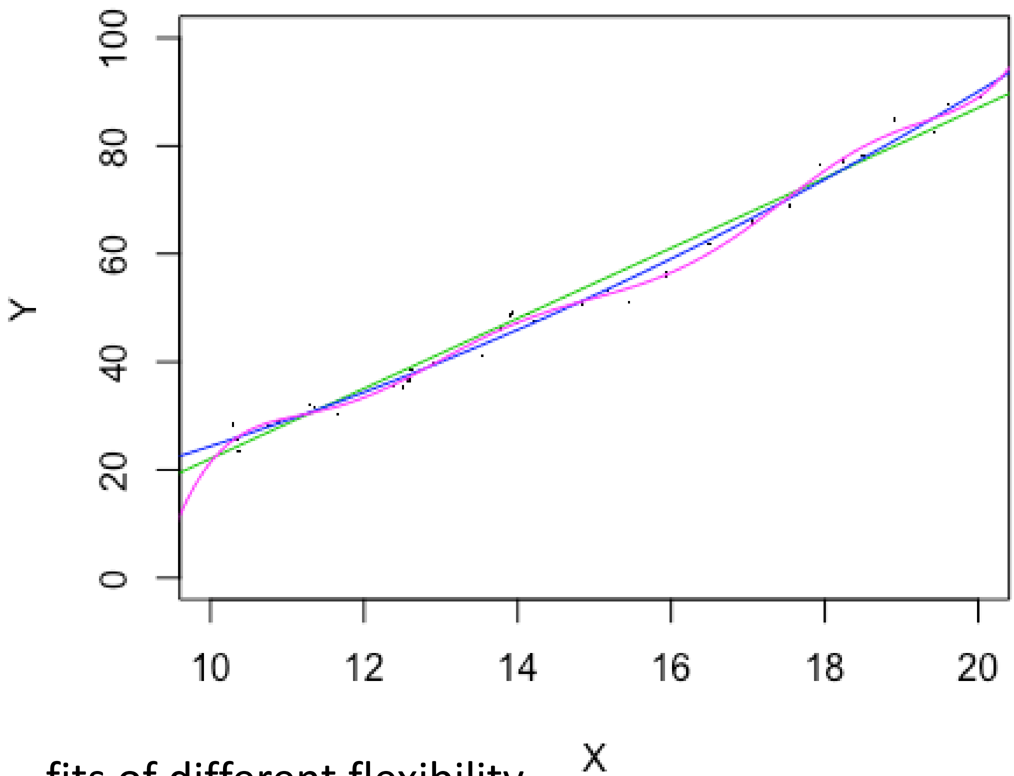
fits of different flexibility

Linear fit

Quadratic fit

10^{th} order polynomial fit

Overfitting



The more flexibility the closer the fit curve can get to the data points.

If the fit function is too flexible, when we optimize it can fit the "noise" component: **OVERFITTED**

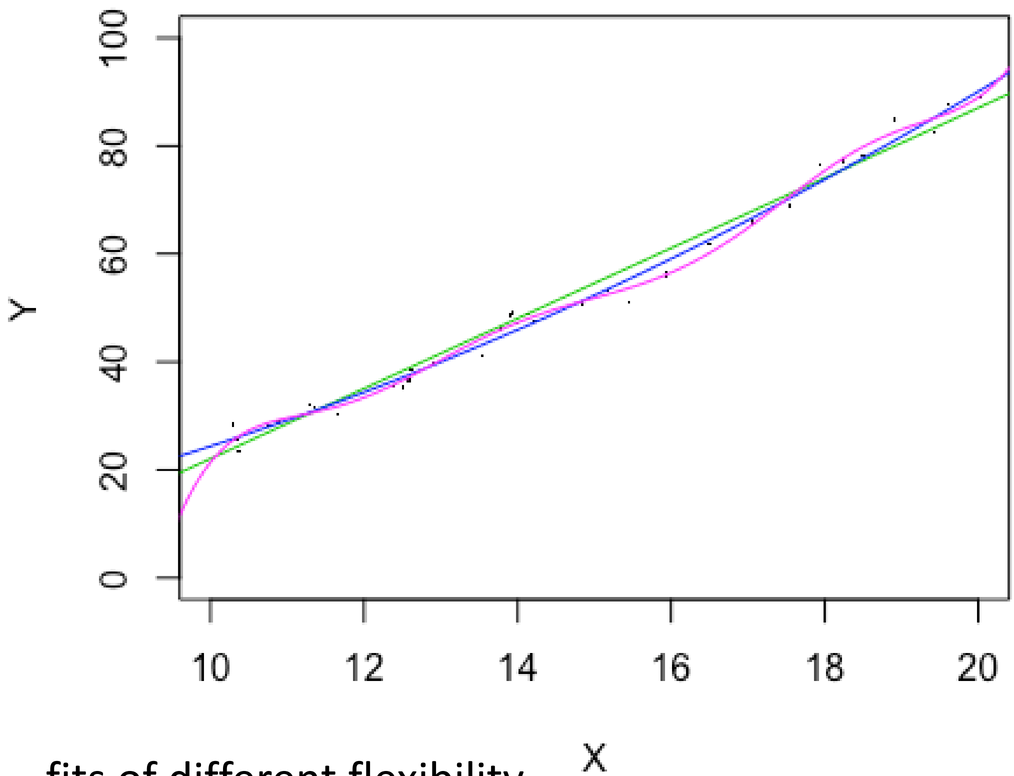
fits of different flexibility

Linear fit

Quadratic fit

10th order polynomial fit

Overfitting



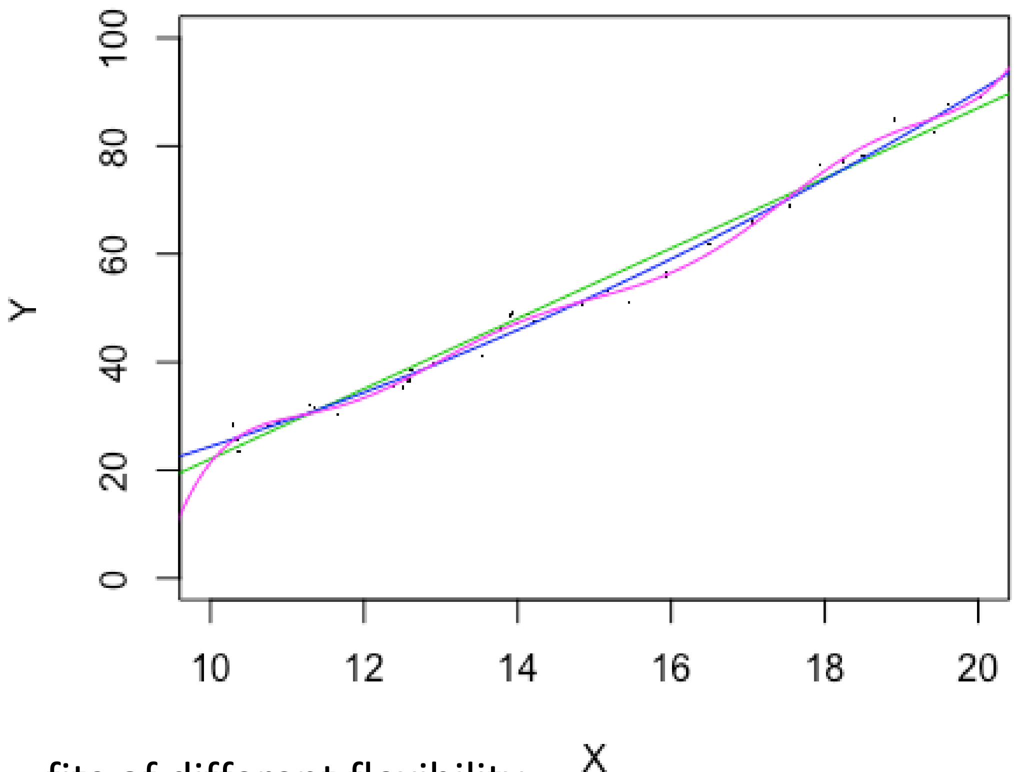
fits of different flexibility
Linear fit
Quadratic fit
10th order polynomial fit

OVERFITTED

- fit function has fitted the "noise" in the training data.
- Makes it a worse estimate of the true relationship.
- Shows smaller residuals for the training data.
- But may perform poorly on new data where the "noise" on points differs.

If you see much worse fit performance on new data compared to the training data you may have overfitted.

Overfitting



fits of different flexibility
Linear fit
Quadratic fit
10^{th} order polynomial fit

OVERFITTING

Factors that can lead to overfitting:

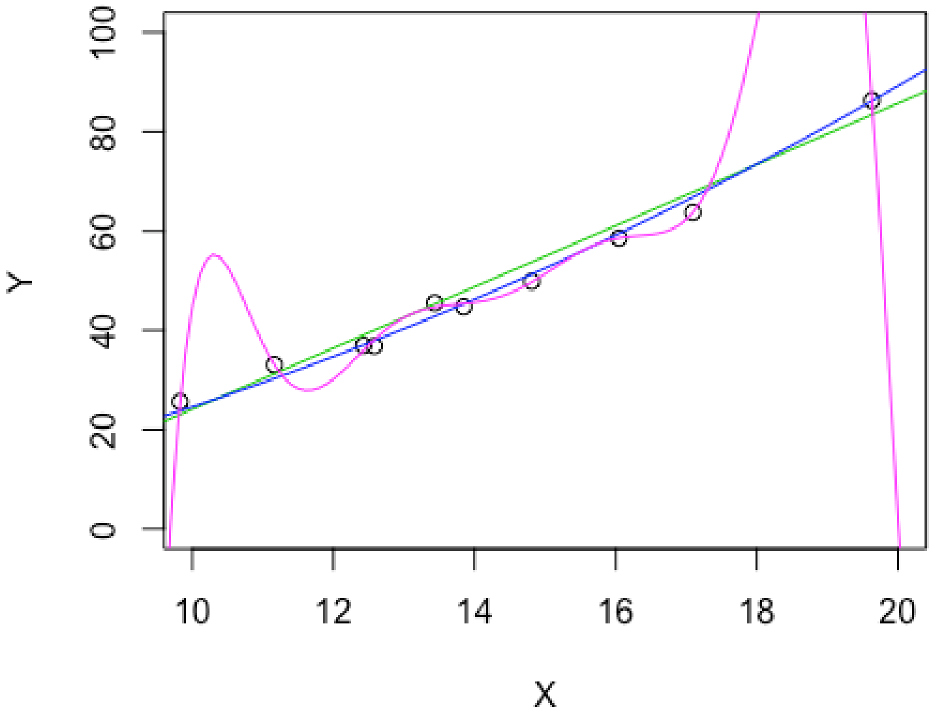
- fit function is more flexible than the true regression function
- few data points

(With lots of data the noise on individual points is less able to influence the fit.)

Overfitting

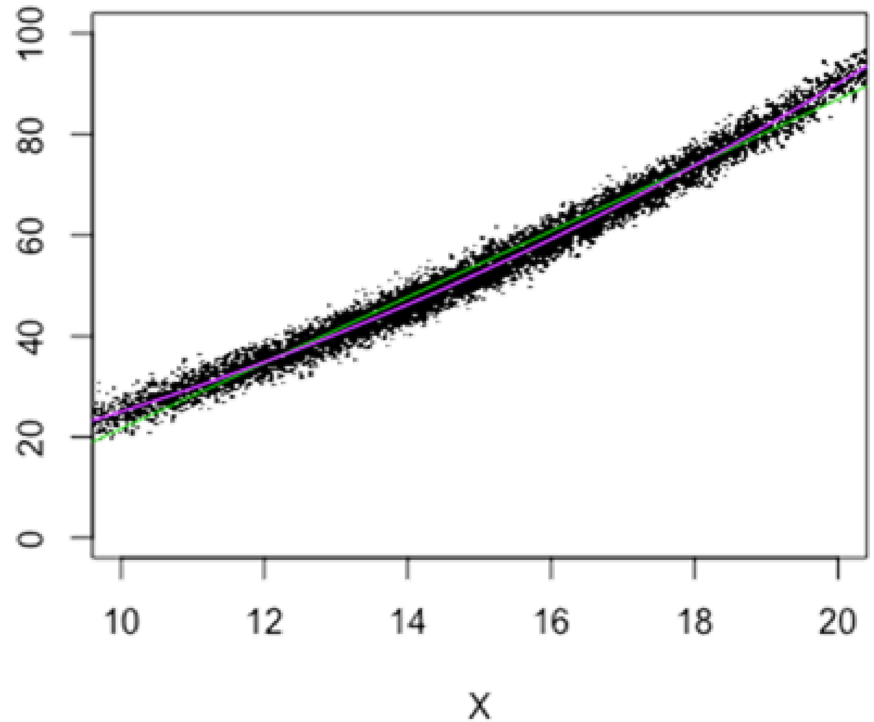
very few data points

Flexible fit gets close to each data point used to train



many data points

Flexible fit gives good match to true regression relationship



fits of different flexibility

Linear fit

Quadratic fit

10th order polynomial fit

Too inflexible a fit function:

Fit function can't match true relationship.

(e.g. will not get optimal performance, even if we estimate the regression function using many data points).

Too flexible a fit function:

Fit function may overfit and be influenced by the noise in our training data.

(e.g. will not get optimal performance because we fail to get a good estimate for the true regression relationship).

Too inflexible a fit function:

Fit function can't match true relationship.

(e.g. will not get optimal performance, even if we estimate the regression function using many data points).

*our fit has **bias** – even estimates on large data sets will give residuals that are too big.*

Too flexible a fit function:

Fit function may overfit and be influenced by the noise in our training data.

(e.g. will not get optimal performance because we fail to get a good estimate for the true regression relationship).

*our fit has **variability** – it is fitting to the noise, so our estimated function will vary according to the sample used to train*

Too inflexible a fit function:

Fit function can't match true relationship.

(e.g. will not get optimal performance, even if we estimate the regression function using many data points).

*our fit has **bias** – even estimates on large data sets will give residuals that are too big. We have **underfitted.***

Too flexible a fit function:

Fit function may overfit and be influenced by the noise in our training data.

(e.g. will not get optimal performance because we fail to get a good estimate for the true regression relationship).

*our fit has **variability** – we have **overfitted** and fit is too influenced by noise in data. Our estimated function will vary according to the data sample used to train.*

bias – variability trade off

bias – variability trade off

Our example used polynomial fits – higher order polynomial meant more flexibility in the fit.

bias – variability trade off

Our example used polynomial fits – higher order polynomial meant more flexibility in the fit.

However in general the flexibility of a multivariate fit depends on the number of fit coefficients (more coefficients to vary = more flexibility)

Therefore the more predictors we include the more flexible our fit is overall.

bias – variability trade off

Our example used polynomial fits – higher order polynomial meant more flexibility in the fit.

However in general the flexibility of a multivariate fit depends on the number of fit coefficients (more coefficients to vary = more flexibility)

Therefore the more predictors we include the more flexible our fit is overall.

Adding genuine predictors can improve fit function by reducing bias.

better fit

Adding predictors will also increase the risk of overfitting.

(even if the predictor is genuine)

poorer fit

bias – variability trade off

Our example used polynomial fits – higher order polynomial meant more flexibility in the fit.

However in general the flexibility of a multivariate fit depends on the number of fit coefficients (more coefficients to vary = more flexibility)

Therefore the more predictors we include the more flexible our fit is overall.

Adding genuine predictors can improve fit function by reducing bias.

better fit

Adding predictors will also increase the risk of overfitting.

(even if the predictor is genuine)

poorer fit

We need a way to detect overfitting and optimize the trade off between bias and variability…..

Diagnosing overfitting

Overfitting means we have fitted the noise in the data used to train the model.

High risk of overfitting if $\mathbf{p} \approx \mathbf{n}$

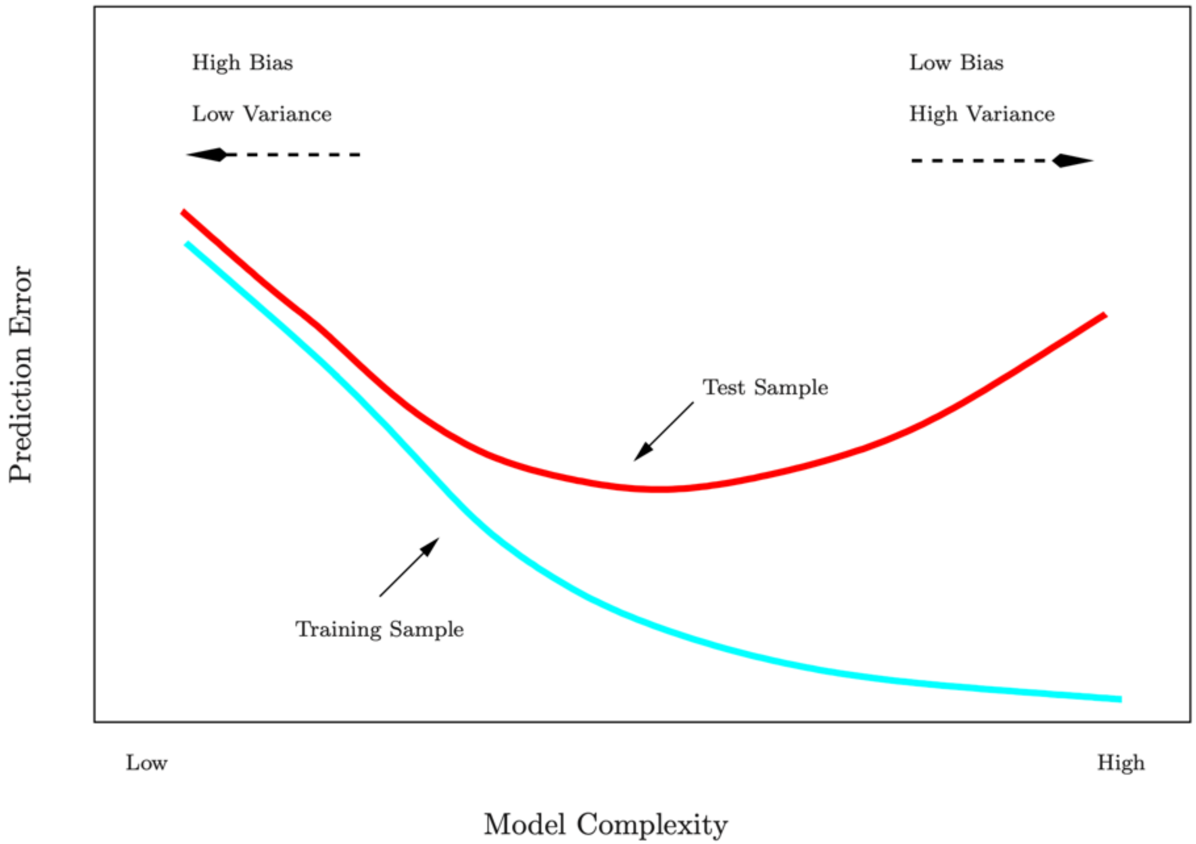
n – number of data points   p - number of predictor coefficients

An overfitted model will therefore give better performance on the training data than data that was not used for training.

--> Compare performance between a training data set and a test data set, if they are close then we have not overfitted.

Choosing a fit function

An overfitted model will therefore give better performance on the training data than data that was not used for training.



--> Compare performance between a training data set and a test data set, if they are close then we have not overfitted.

Optimising our fit to training data

We often have choices:

e.g. what fit function to use?
which predictors to include?
what predictor transformations (e.g. log) might be useful?

Look at performance of fit (i.e. minimize residuals) for each option.
Choose the best performing fit.

Optimising our fit to training data

We often have choices:

e.g. what fit function to use?
what predictors to include?
what predictor transformations (e.g. log) might be useful?

Look at performance of fit (i.e. minimize residuals) for each option.
Choose the best performing fit.

PROBLEM: Performance on training data ≠ real world performance.
We may not choose the best fitting model.

Optimising our fit to training data

We often have choices:

e.g. what fit function to use?
what predictors to include?
what predictor transformations (e.g. log) might be useful?

Look at performance of fit (i.e. minimize residuals) for each option.
Choose the best performing fit.

PROBLEM: Performance on training data ≠ real world performance.
We may not choose the best fitting model.

SOLUTION 1: Use measures of fit that account for overfitting:
(Adj Rsquared, AIC, BIC)

Optimising our fit to training data

We often have choices:

e.g. what fit function to use?
what predictors to include?
what predictor transformations (e.g. log) might be useful?

Look at performance of fit (i.e. minimize residuals) for each option.
Choose the best performing fit.

PROBLEM: Performance on training data ≠ real world performance.
We may not choose the best fitting model.

SOLUTION 1: Use measures of fit that account for overfitting:
(Adj Rsquared, AIC, BIC)

ISSUE: These are accurate only if linear fits assumptions are valid

Optimising our fit to training data

We often have choices:

e.g. what fit function to use?
what predictors to include?
what predictor transformations (e.g. log) might be useful?

Look at performance of fit (i.e. minimize residuals) for each option.
Choose the best performing fit.

PROBLEM: Performance on training data ≠ real world performance.
We may not choose the best fitting model.

SOLUTION 2: validate performance measurements by testing on
(more general) data that was not used to train the fit.

Optimising our fit to training data

We often have choices:

e.g. what fit function to use?
what predictors to include?
what predictor transformations (e.g. log) might be useful?

Look at performance of fit (i.e. minimize residuals) for each option.
Choose the best performing fit.

PROBLEM: Performance on training data ≠ real world performance.
We may not choose the best fitting model.

SOLUTION 2: validate performance measurements by testing on
(more general) data that was not used to train the fit.

HOW: Split some data from the available full dataset and use it for testing performance after model has been trained.

Optimising our fit to training data

We often have choices:

e.g. what fit function to use?
what predictors to include?
what predictor transformations (e.g. log) might be useful?

Look at performance of fit (i.e. minimize residuals) for each option.
Choose the best performing fit.

PROBLEM: Performance on training data ≠ real world performance.
We may not choose the best fitting model.

SOLUTION 3: use a cross-validation method
(more general)

(we will cover these later)