ORF307 – Optimization

4. Least squares data fitting

Bartolomeo Stellato — Spring 2025

Today’s lecture

Least squares data fitting

- Least squares model fitting
- Univariate regression
- Multivariate regression
- Validation
- Example

Least squares model fitting

Setup

We believe a scalar y and a n-vector are related by a model

$$y \approx f(x)$$

- x is the *independent variable* or *feature vector*
- y is the *outcome* or *response variable*
- $f : \mathbf{R}^n \to \mathbf{R}$ maps x to y

We don't know f and we want to estimate it from data

Data

All we have is data

n-vectors $x^{(1)}, \ldots, x^{(N)}$, and scalars $y^{(1)}, \ldots, y^{(N)}$

also called *observations, examples, samples* or *measurements*.

$(x^{(i)}, y^{(i)})$ is the ith data pair

$x_j^{(i)}$ is the jth component of ith data point $x^{(i)}$

Model

Guess a model $\hat{f} : \mathbf{R}^n \to \mathbf{R}$ to approximate f

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

- $f_i : \mathbf{R}^n \to \mathbf{R}$ are *feature mappings* or *basis functions*
- θ_i are *model parameters* to choose
- $\hat{y}^{(i)} = \hat{f}(x^{(i)})$ is the model's *prediction* for $y^{(i)}$
- Goal: $\hat{y}^{(i)} \approx y^{(i)}$ (consistent with observed data)

Least squares data fitting

Prediction error (residual) $r^{(i)} = y^{(i)} - \hat{y}^{(i)}$

Goal: choose model parameters θ_i to minimize mean squared error (MSE)

$$\frac{(r^{(1)})^2 + \cdots + (r^{(N)})^2}{N}$$

This can be formulated as a least squares problem

Note. we sometimes compute the root mean squared error RMS = $\sqrt{\text{MSE}}$ because it has the same units as $y^{(i)}$

Least squares data fitting

Vector form

Express problems with N-vectors

- $y^{\mathrm{d}} = (y^{(1)}, \dots, y^{(N)})$, vector of outcomes
- $\hat{y}^{\mathrm{d}} = (\hat{y}^{(1)}, \dots, \hat{y}^{(N)})$, vector of predictions
- $r^{\mathrm{d}} = (r^{(1)}, \dots, r^{(N)})$, vector of residuals

$\longrightarrow$

Goal

minimize $\quad \|r^{\mathrm{d}}\|^2$

We can write $\hat{y}^{(i)} = \hat{f}(x^{(i)})$ in terms of parameters θ_i

$$\hat{y}^{(i)} = A_{i1}\theta_1 + \cdots + A_{ip}\theta_p, \qquad A_{ij} = f_j(x^{(i)}) \qquad \longrightarrow \qquad \hat{y}^{\mathrm{d}} = A\theta$$

Least squares problem

$$\text{minimize} \quad \|r^{\mathrm{d}}\|^2 = \|y^{\mathrm{d}} - \hat{y}^{\mathrm{d}}\|^2 = \|y^{\mathrm{d}} - A\theta\|^2 = \|A\theta - y^{\mathrm{d}}\|^2$$

Solution

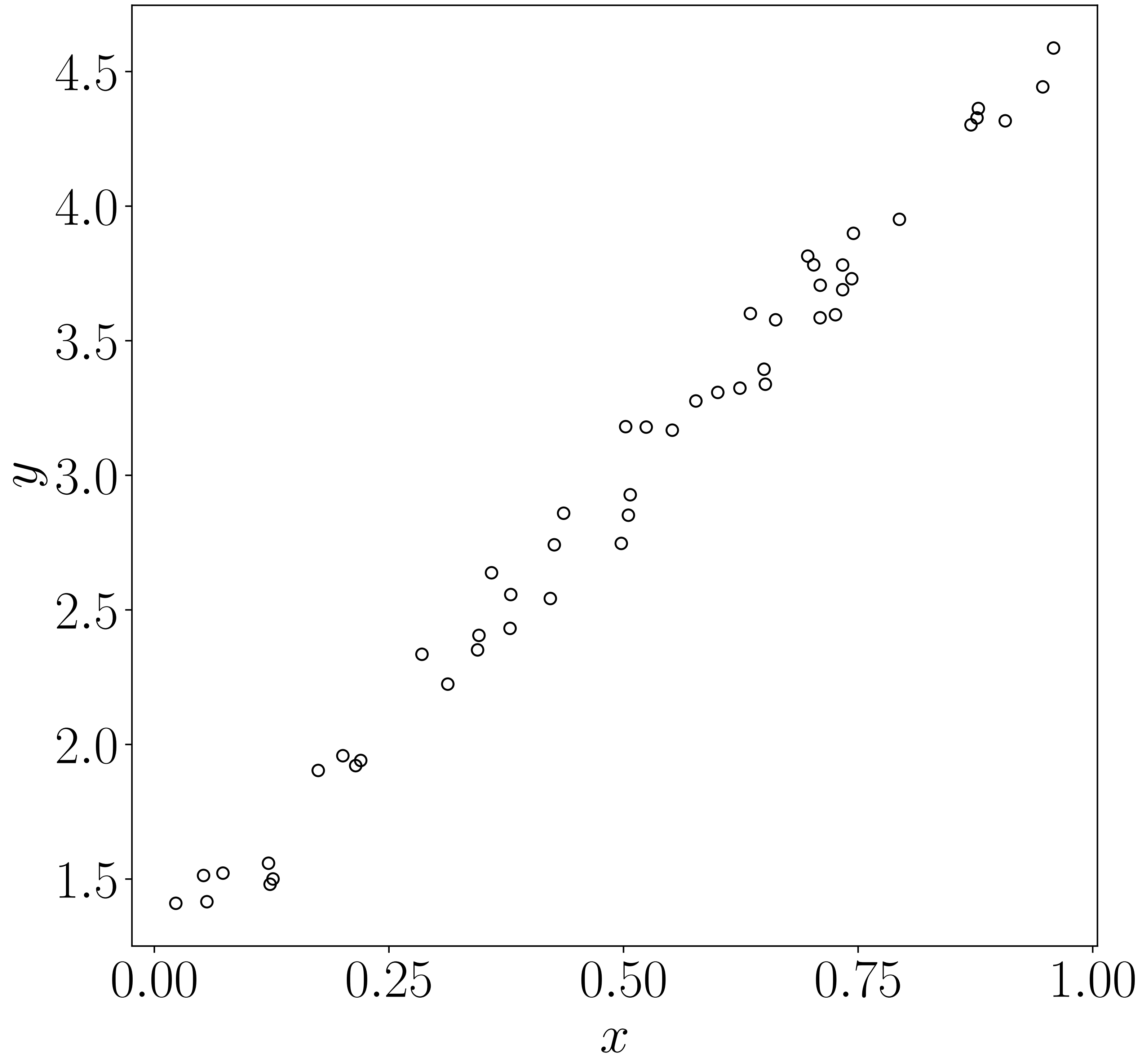
$$(A^T A)\theta^\star = A^T y^{\mathrm{d}}$$

Univariate fitting

Fitting univariate functions

We seek to approximate a function $f : \mathbf{R} \to \mathbf{R}$ $\qquad (n = 1)$

Data points $(x^{(i)}, y^{(i)})$



Straight line fit

Model

$$\hat{f}(x) = \theta_1 + \theta_2 x$$

- Parameters: $\theta = (\theta_1, \theta_2) \qquad (p = 2)$
- Features: $f_1(x) = 1,\ f_2(x) = x$

Least squares data

$$A = \begin{bmatrix} 1 & x^{(1)} \\ 1 & x^{(2)} \\ \vdots & \vdots \\ 1 & x^N \end{bmatrix} = \begin{bmatrix} \mathbf{1} & x^{\mathrm{d}} \end{bmatrix}$$

Goal

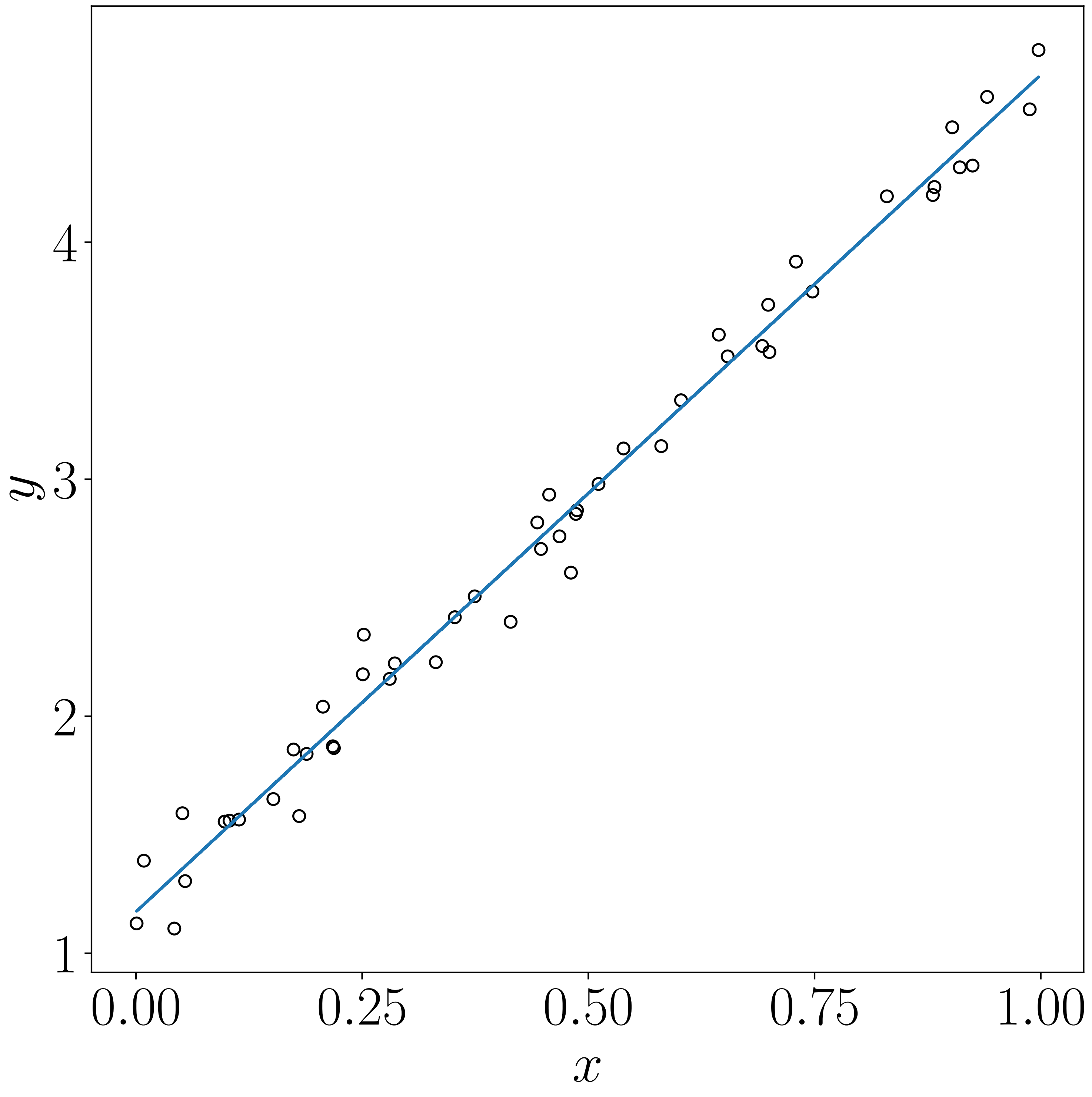
minimize $\quad \|r^{\mathrm{d}}\|^2 = \|A\theta - y^{\mathrm{d}}\|^2$

Straight line fit

Example

$$\theta^\star = (1.75, 3.53)$$

$$\hat{f}(x) = \theta_1 + \theta_2 x$$



Asset α and β in finance

whole market returns

$$x^{\mathrm{d}} = (r_1^{\mathrm{mkt}}, \ldots, r_T^{\mathrm{mkt}})$$

r_t^{mkt} is the return of the *whole market* at time t

individual asset returns

$$y^{\mathrm{d}} = (r_1^{\mathrm{ind}}, \ldots, r_T^{\mathrm{ind}})$$

r_t^{ind} is the return of an *individual asset* at time t

→

Goal
predict individual asset return from whole market return

Linear model

$$\hat{y} = (r^{\mathrm{rf}} + \alpha) + \beta(x - \mu^{\mathrm{mkt}})$$

asset α
average asset return above r^{rf}

asset β
relates market return fluctuations to asset return

- μ^{mkt} is the average market return over period
- r^{rf} is the risk-free interest rate over the period

Time series trend

$y^{(i)}$ is the value of quantity at time $x^{(i)} = i$

$y^{\mathrm{d}} = (y^{(1)}, \ldots, y^{(N)})$ is the *time series*

Model (trend line)

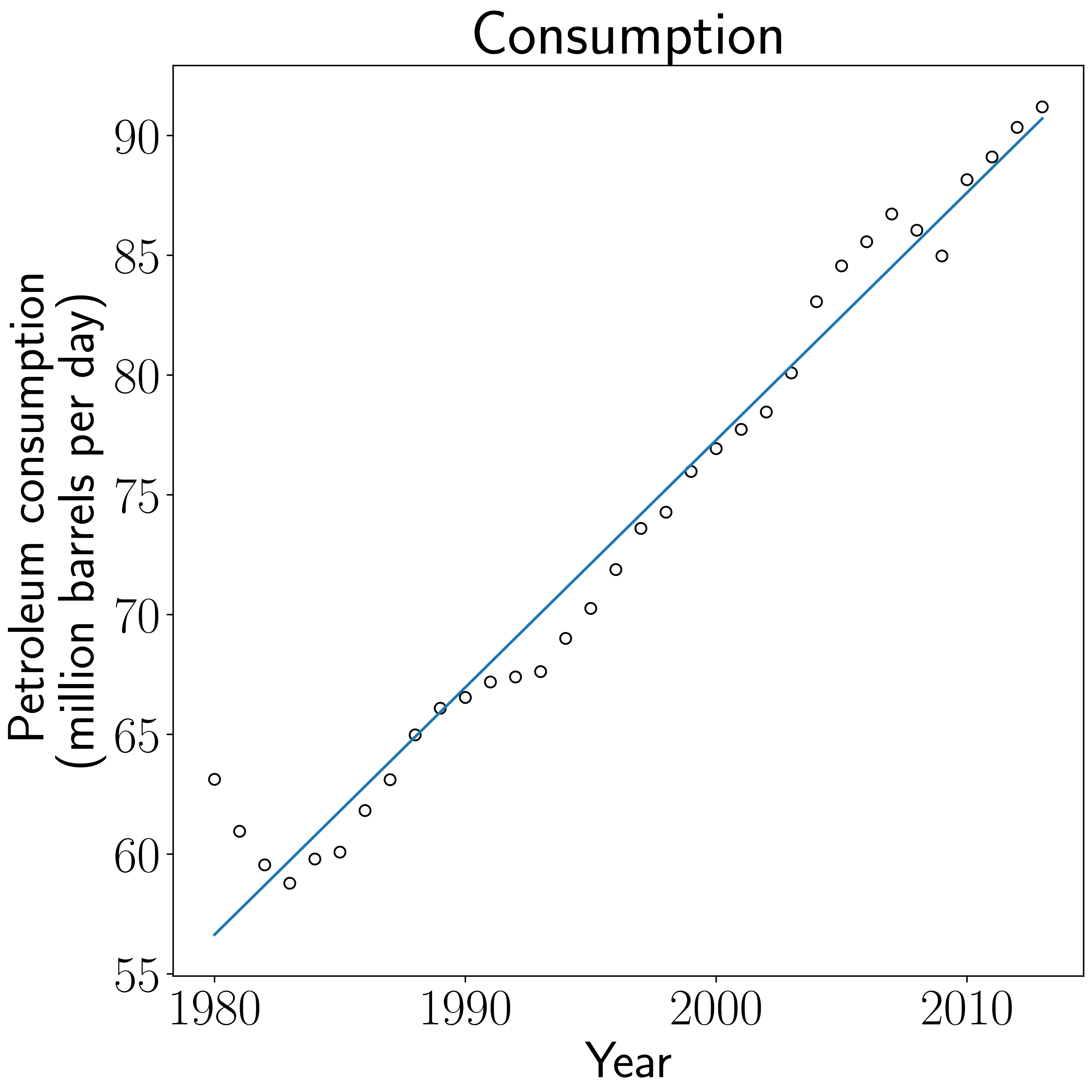
$$\hat{y}^{(i)} = \theta_1 + \theta_2 i, \qquad i = 1, \ldots, N$$

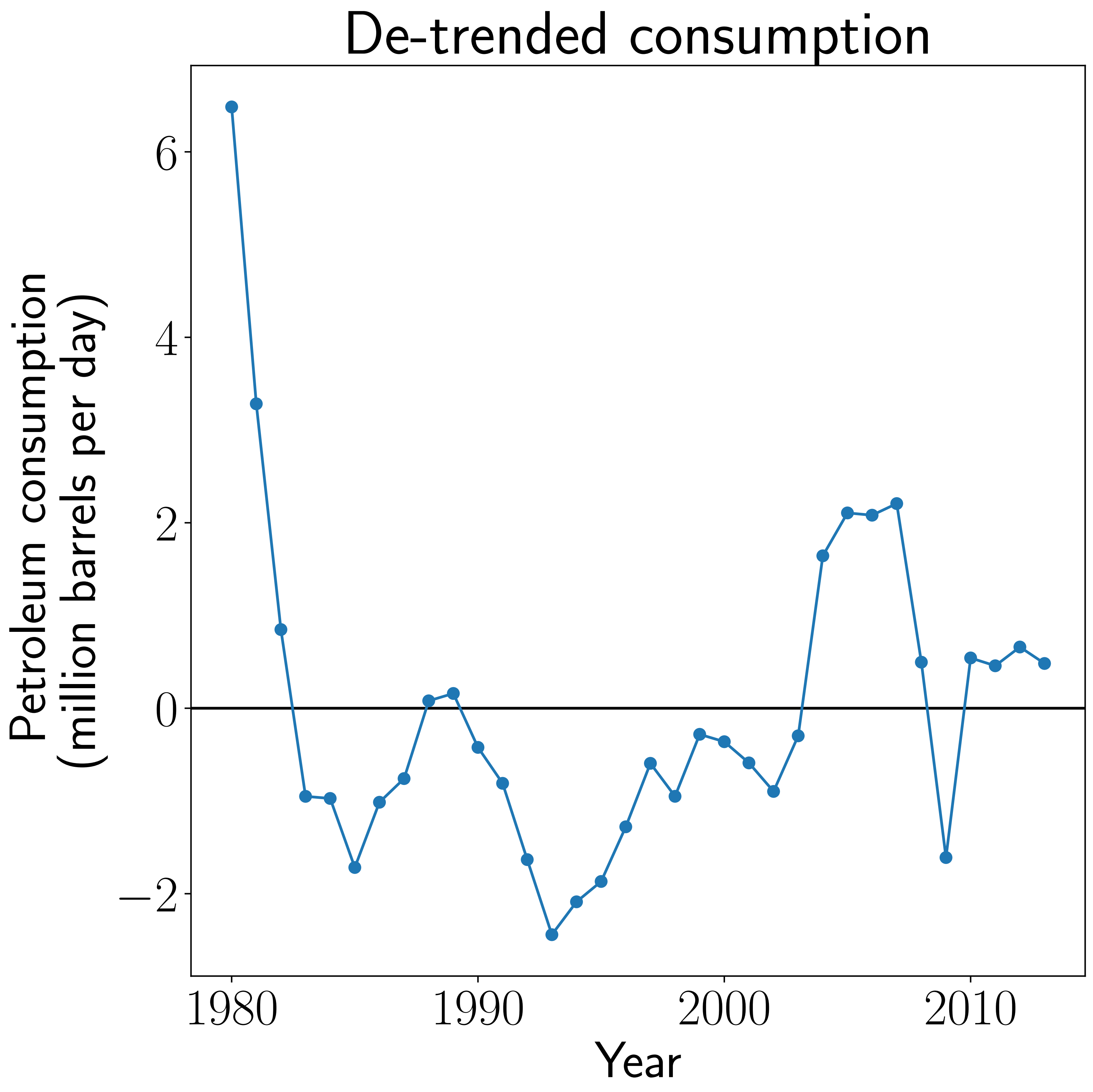
trend coefficient (θ_2)

$y^{\mathrm{d}} - \hat{y}^{\mathrm{d}}$ is the *de-trended time series*

Time series trend

Petroleum consumption





Polynomial fit

Features

$$f_i(x) = x^{i-1}, \quad i = 1, \dots, p$$

Model

$$\hat{f}(x) = \theta_1 + \theta_2 x + \dots \theta_p x^{p-1}$$

degree at most $p - 1$

Least squares data

Vandermonde matrix

$$A = \begin{bmatrix} 1 & x^{(1)} & \dots & (x^{(1)})^{p-1} \\ 1 & x^{(2)} & \dots & (x^{(2)})^{p-1} \\ \vdots & \vdots & & \vdots \\ 1 & x^{(N)} & \dots & (x^{(N)})^{p-1} \end{bmatrix}$$

Notation remark

x^i means scalar to ith power

$x^{(i)}$ means ith data point

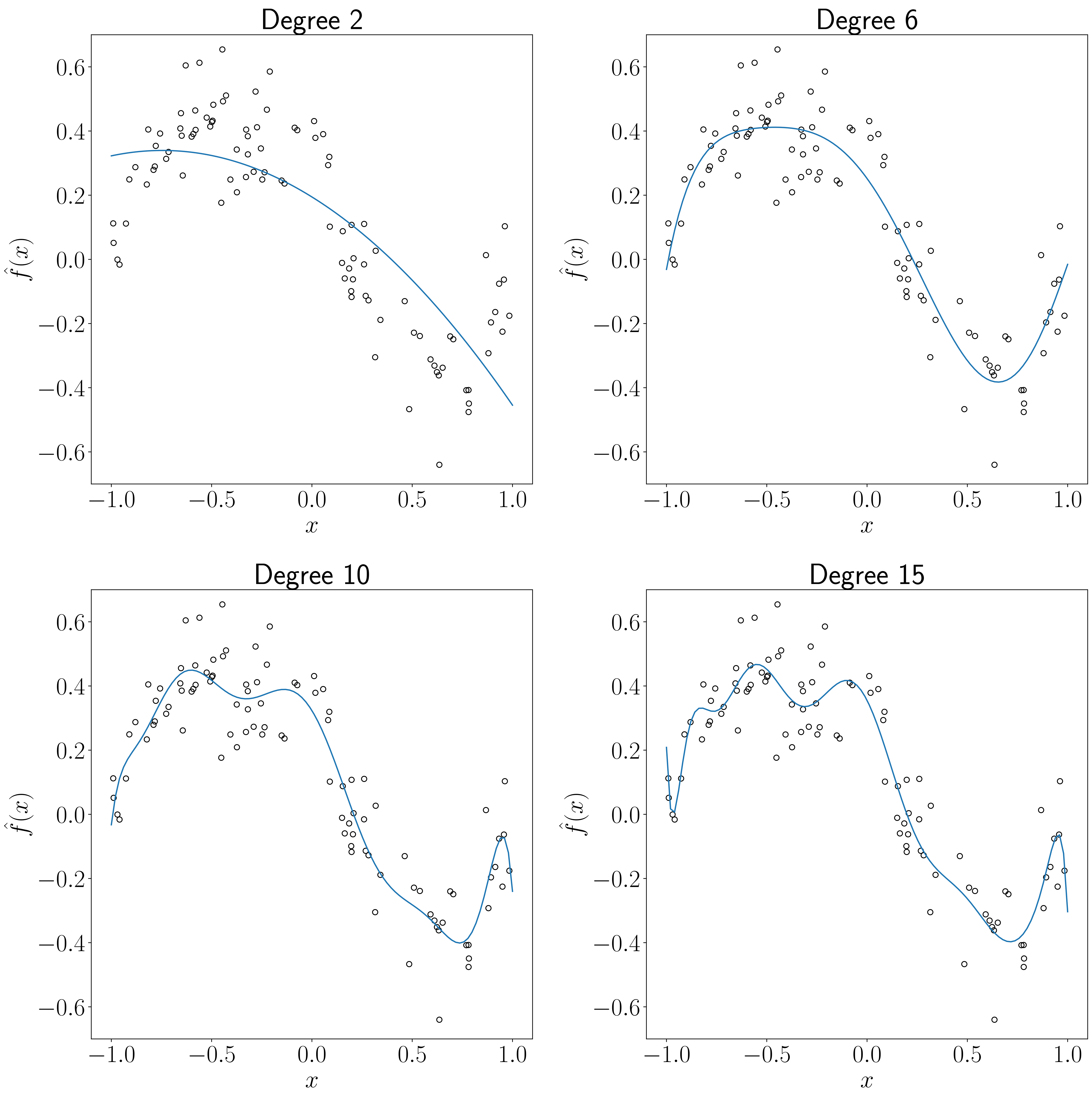
Goal

minimize $\quad \|r^{\mathrm{d}}\|^2 = \|A\theta - y^{\mathrm{d}}\|^2$

Polynomial fit

$N = 100$ data points

Which model is better?



Auto-regressive time series model

$z_1, z_2, \ldots$ is a **time series**

auto-regressive (AR) prediction model

$$\hat{z}_{t+1} = \theta_1 z_t + \cdots + \theta_M z_{t-M+1}, \quad t = M, M+1, \ldots$$

(predict $\hat{z}_{t+1}$ based on previous M values, where M is the memory)

Goal: Chose θ to minimize sum of squares of prediction errors

$$(\hat{z}_{M+1} - z_{M+1})^2 + \cdots + (\hat{z}_T - z_T)^2$$

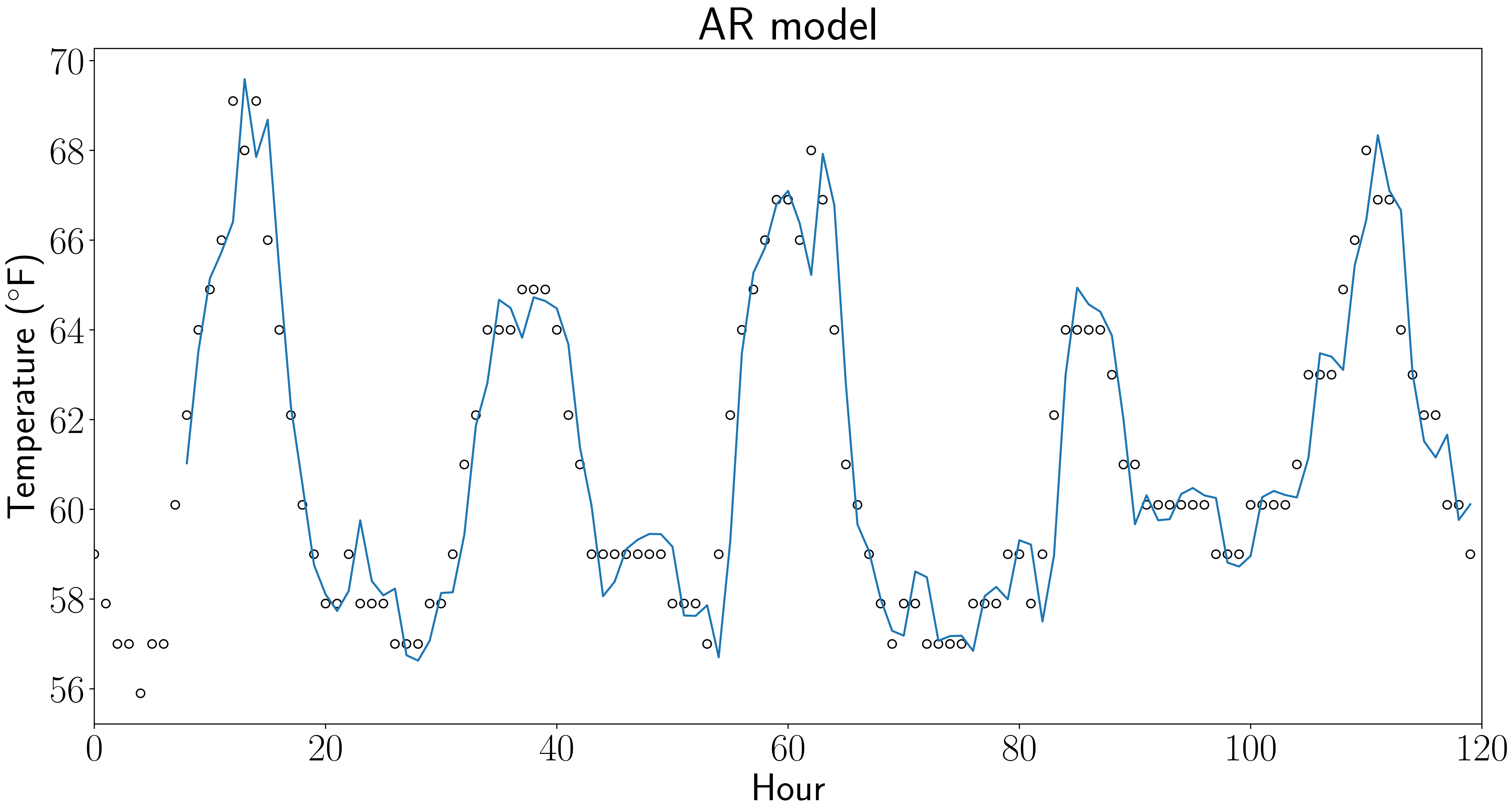
General data fitting form

$$y^{(i)} = z_{M+i}, \qquad x^{(i)} = (z_{M+i-1}, \ldots, z_i), \quad i = 1, \ldots, T-M$$

Auto-regressive time series model

5 days hourly temperature at Los Angeles International Airport (LAX)

- Previous hour: $\hat{z}_{t+1} = z_t$, MSE 1.35
- 24 hours before: $\hat{z}_{t+1} = z_{t-23}$, MSE = 3.00
- AR model with $M = 8$, MSE = 1.02



Multivariate regression

Linear regression as general data fitting

Standard linear regression model

$$\hat{y} = \hat{f}(x) = x^T \beta + v$$

Equivalent general data fitting model

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

with basis functions

$$f_1(x) = 1, \quad f_i(x) = x_{i-1}, \qquad i = 2, \ldots, n-1$$

Therefore, we can write the linear regression model as

$$\hat{y} = \hat{f}(x) = \theta_1 + \theta_2 x_1 + \cdots + \theta_{n+1} x_n = x^T \theta_{2:n} + \theta_1$$

with $\beta = \theta_{2:n+1},\ v = \theta_1$

General data fitting as linear regression

General data fitting model

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

Equivalent linear regression model

$$\hat{y} = \hat{f}(x) = \tilde{x}^T \beta + v$$

- $f_1(x) = 1$ (common assumption)
- $\tilde{x} = (f_2(x), \ldots, f_p(x))$ (transformed features)
- $v = \theta_1$ and $\beta = \theta_{2:p}$ (linear regression parameters)

Our general data fitting framework is nothing more than
linear regression on transformed data

Validation

Generalization

Main goal

- The goal of model is not to predict outcome for *given data*
- Instead, it is to predict the outcome on *new, unseen data*

↓

Seen/Unseen data

- A model that makes reasonable predictions on new, unseen data has **generalization ability** or it **generalizes**

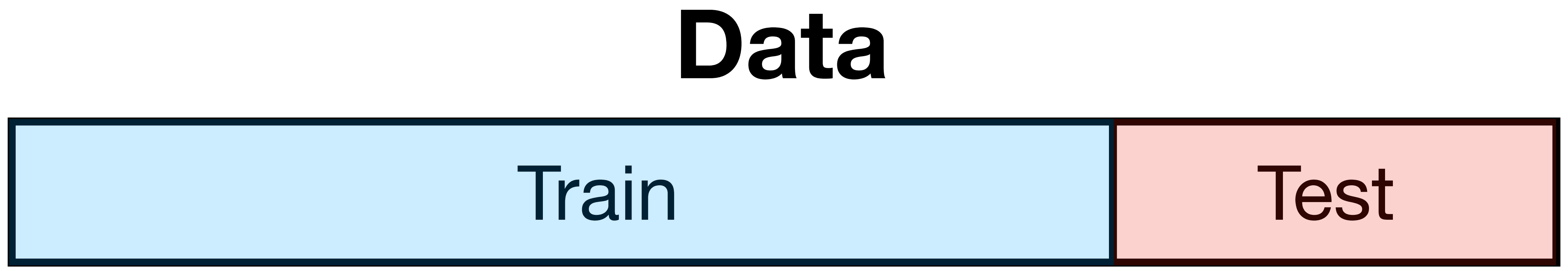
- A model that makes poor predictions on new, unseen data is said to suffer from **over-fitting**

(Almost) always true in decision making
The objective function (here, the training error) is just a "surrogate" of the true goal

Overfitting example

Validation

Simple and effective method to guess if a model generalize



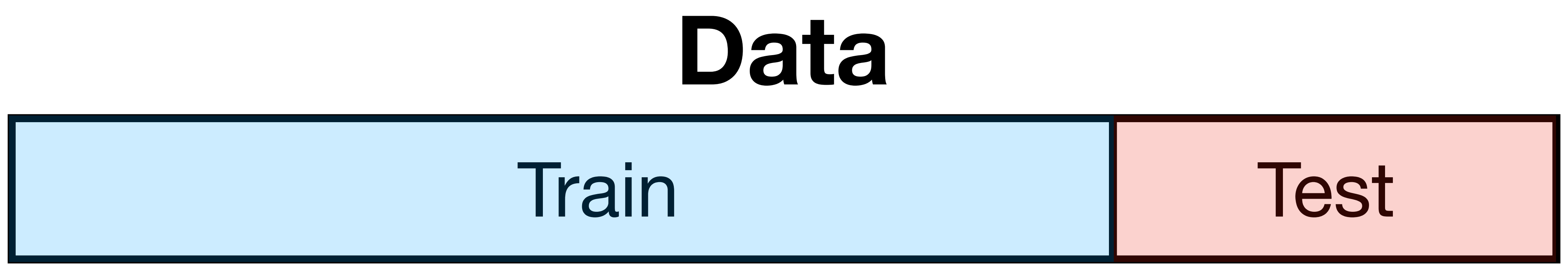
1. Split data in *training* and *test set* (typical 80%/20% or 90%/10%)
2. Build (train) model on training data set (i.e., compute $\theta^\star$)
3. Check model's prediction on test data set

Compare the MSE prediction error on test vs train data set

↓

If similar, we can *guess* that the model will generalize

Validation



Useful to choose among different candidate models

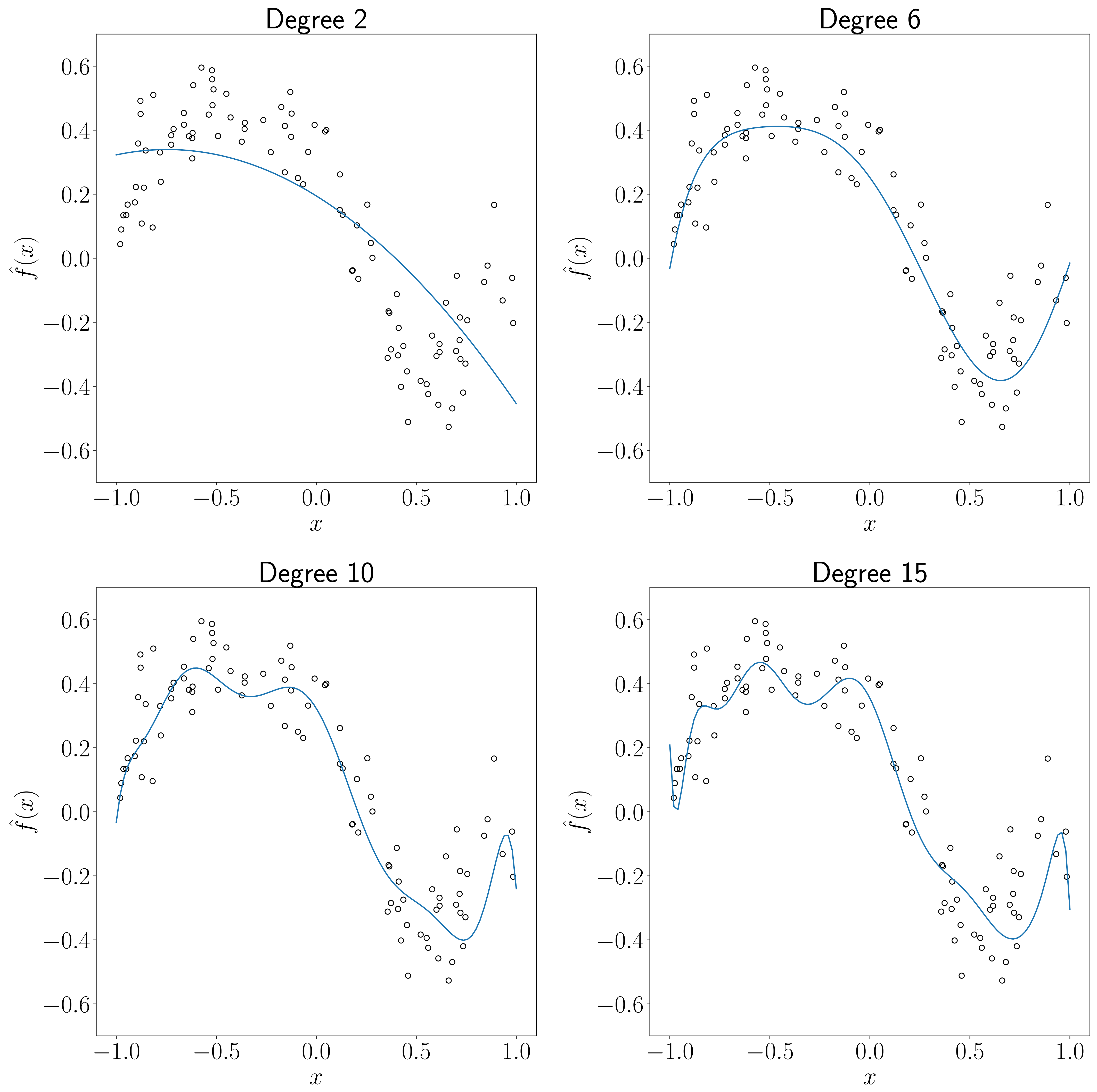
- Polynomials of different degrees
- Models with different transformed features

↓

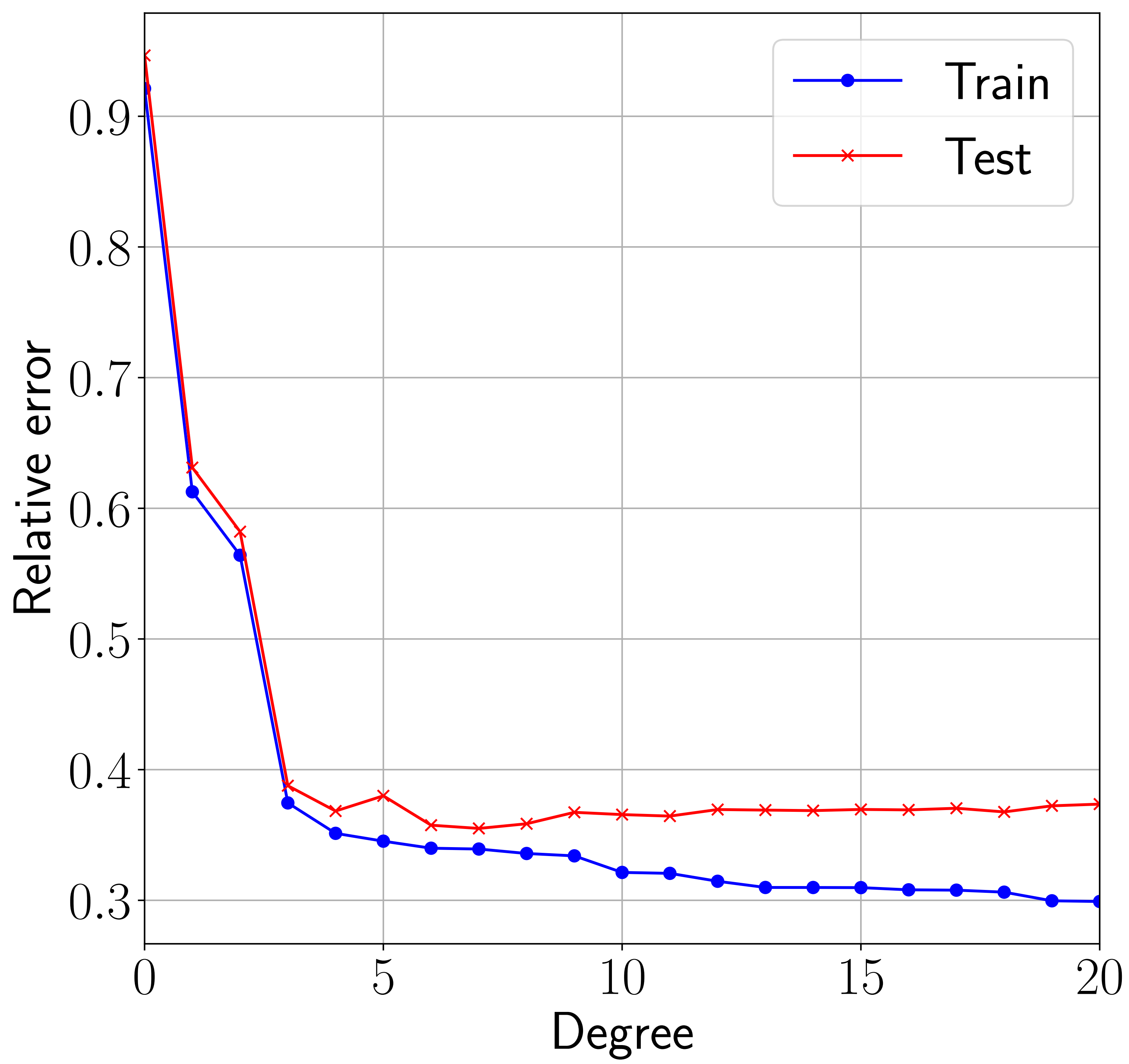
We choose the one with ***lowest test error***

Example

Polynomial fit



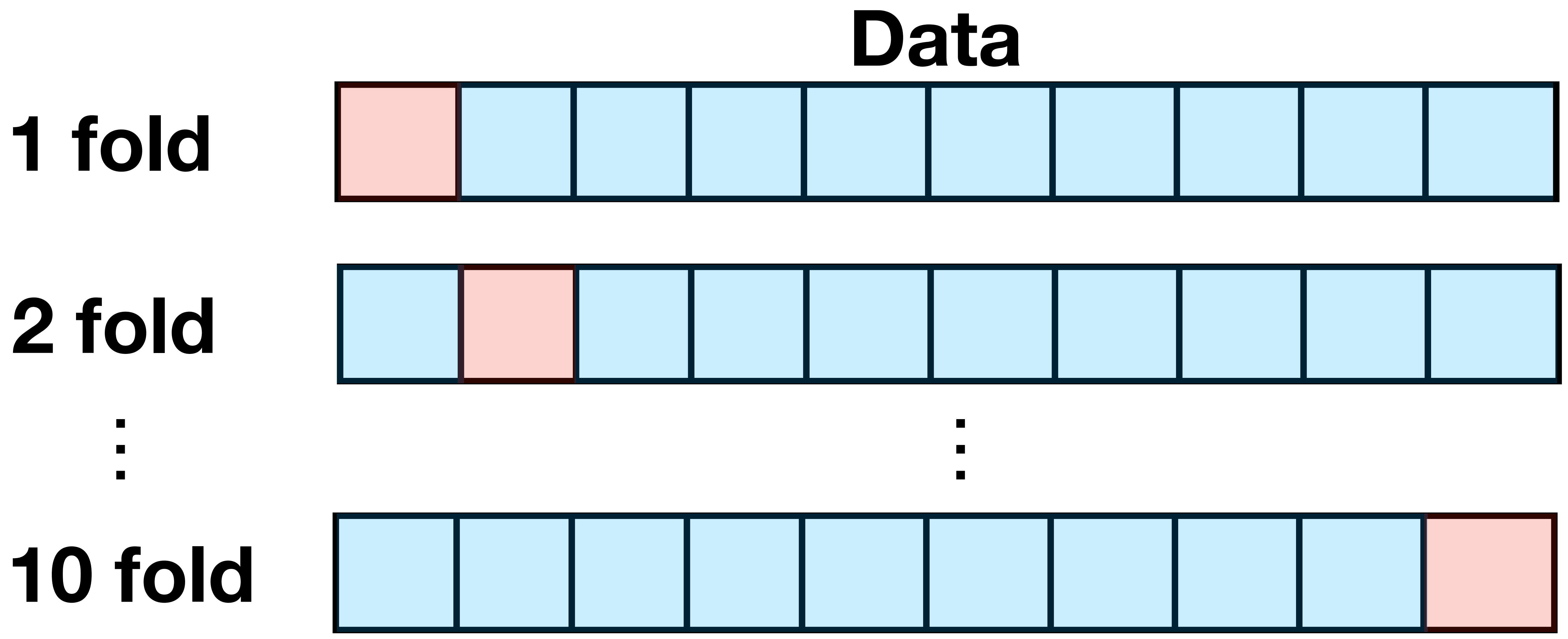
100 points *training set*,
100 points *test set*



It suggests degrees 6/7/8 are reasonable choices

Cross validation

Check **which method** provides the better models
(e.g., choice of features, basis functions, etc.)



1. Divide data in 10 folds
2. For $i = 1, \ldots, 10$ build (train) model using all folds except i
3. Test model on data in fold i

For each fold, compare the MSE prediction error on test vs train data

↓

If test vs train MSE are similar and consistent, we can *guess* the model will generalize

Remark

This returns many models, to get a model, we have to afterwards train over the whole data set

Example

House price regression

774 house sales in Sacramento area

Model

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_8 f_8(x)$$

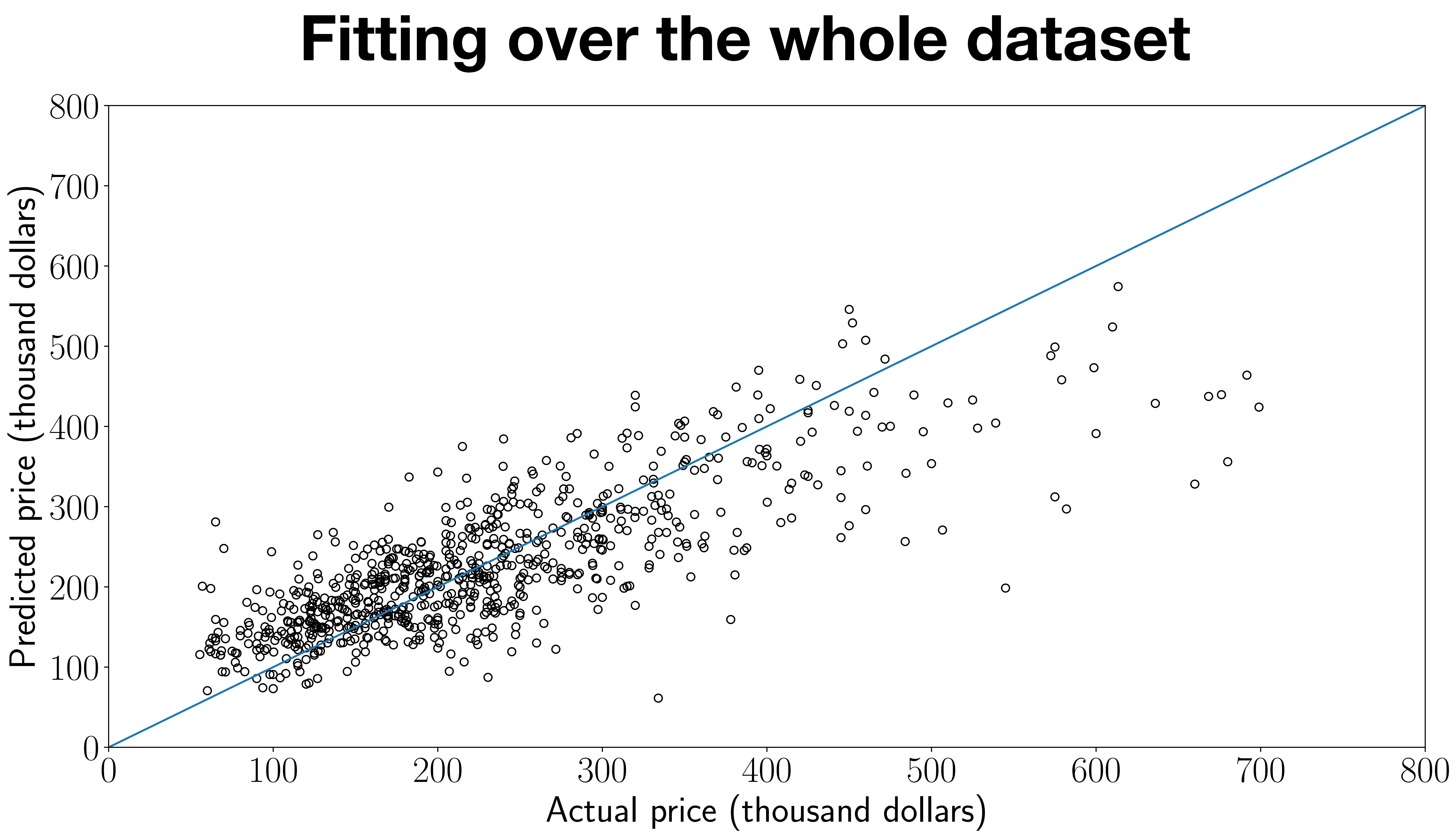
Base features

- x_1 is the area of the house (in $1000\mathrm{ft}^2$)
- x_2 is the number of bedrooms
- x_3 is 1 for condo and 0 for house (boolean)
- x_4 is the ZIP code (62 values)

Transformed features

- $f_1(x) = 1$ (offset)
- $f_2(x) = x_1$
- $f_3(x) = \max\{x_1 - 1.5, 0\}$ is the house area above $1500\mathrm{ft}^2$
- $f_4(x) = x_2$
- $f_5(x) = x_3$
- $f_6(x), f_7(x), f_8(x)$ are Boolean functions of x_4 to encode 4 groups of nearby zip codes (i.e., neighborhood)

House price prediction



How can we be sure it generalizes?

House price regression

Crossvalidation

5 folds of 155 sales each

Fold	Train error	Test error	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8
1	0.26	0.30	115.60	177.53	-47.05	-14.64	-13.90	-112.00	-122.59	-36.51
2	0.26	0.29	121.59	165.48	-30.35	-17.74	-20.21	-95.22	-103.67	-7.94
3	0.27	0.25	117.83	181.18	-49.78	-19.30	-18.68	-106.50	-112.76	-32.37
4	0.27	0.25	104.00	174.54	-41.96	-18.81	-17.13	-85.68	-92.09	-6.99
5	0.27	0.27	119.40	178.57	-44.82	-19.50	-25.98	-103.95	-111.56	-36.89

Good feature choice

- Models (parameters) reasonably stable across folds
- Similar train and test errors

Least squares data fitting

Today, we learned to:

- **Formulate** many data fitting problems as least squares
- **Avoid** overfitting by keeping our models simple
- **Compare** our models using validation

References

- S. Boyd, L. Vandenberghe: Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares
  - Chapter 13: least squares data fitting

Next lecture

- Multi-objective least squares