9. Least squares data fitting

- model fitting
- regression
- linear-in-parameters models
- time series examples
- validation
- least squares classification
- statistics interpretation

Model fitting

suppose x and a scalar quantity y are related as

$$y \approx f(x)$$

- x is the *explanatory variable* or *independent variable*
- y is the *outcome*, or *response variable*, or *dependent* variable
- we don't know f, but have some idea about its general form

Model fitting

- find an approximate *model* $\hat{f}$ for f, based on observations
- we use the notation $\hat{y}$ for the model *prediction* of the outcome y:

$$\hat{y} = \hat{f}(x)$$

Prediction error

we have data consisting of N *examples* (*samples*, *measurements*, *observations*):

$$x^{(1)}, \ldots, x^{(N)}, \qquad y^{(1)}, \ldots, y^{(N)}$$

- model prediction for example i is $\hat{y}^{(i)} = \hat{f}(x^{(i)})$
- the *prediction error* or *residual* for example i is

$$r^{(i)} = y^{(i)} - \hat{y}^{(i)} = y^{(i)} - \hat{f}(x^{(i)})$$

- the model $\hat{f}$ fits the data well if the N residuals $r^{(i)}$ are small
- prediction error can be quantified using the *mean square error* (MSE)

$$\frac{1}{N}\sum_{i=1}^{N}(r^{(i)})^2$$

the square root of the MSE is the RMS error

Outline

- model fitting
- **regression**
- linear-in-parameters models
- time series examples
- validation
- least squares classification
- statistics interpretation

Regression

we first consider the regression model (page 1.30):

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

- here the independent variable x is an n-vector
- the elements of x are the *regressors*
- the model is parameterized by the weight vector β and the offset (intercept) v
- the prediction error for example i is

$$\begin{aligned} r^{(i)} &= y^{(i)} - \hat{f}(x^{(i)}) \\ &= y^{(i)} - (x^{(i)})^T\beta - v \end{aligned}$$

- the MSE is

$$\frac{1}{N}\sum_{i=1}^{N}(r^{(i)})^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - (x^{(i)})^T\beta - v\right)^2$$

Least squares regression

choose the model parameters v, β that minimize the MSE

$$\frac{1}{N}\sum_{i=1}^{N}\left(v + (x^{(i)})^T\beta - y^{(i)}\right)^2$$

this is a least squares problem: minimize $\|A\theta - y^{\mathrm{d}}\|^2$ with

$$A = \begin{bmatrix} 1 & (x^{(1)})^T \\ 1 & (x^{(2)})^T \\ \vdots & \vdots \\ 1 & (x^{(N)})^T \end{bmatrix}, \qquad \theta = \begin{bmatrix} v \\ \beta \end{bmatrix}, \qquad y^{\mathrm{d}} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

we write the solution as $\hat{\theta} = (\hat{v}, \hat{\beta})$

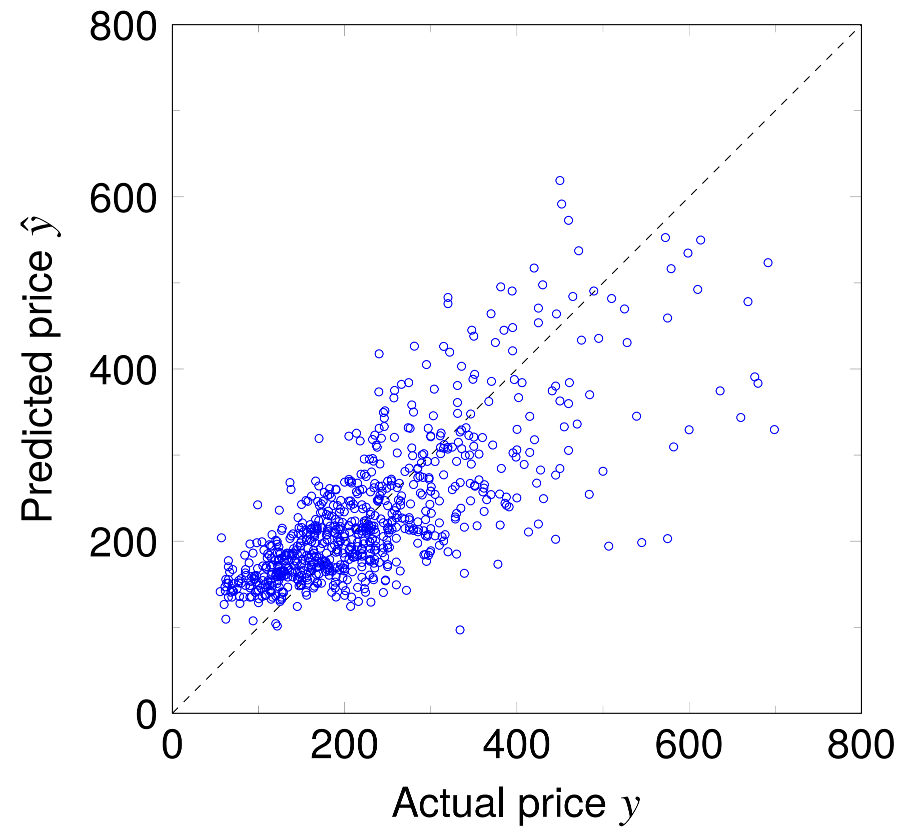
Example: house price regression model

example of page 1.30

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

- $\hat{y}$ is predicted sales price (in 1000 dollars); y is actual sales price
- two regressors: x_1 is house area; x_2 is number of bedrooms

- data set of $N = 774$ house sales
- RMS error of least squares fit is 74.8



Example: house price regression model

regression model with additional regressors

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

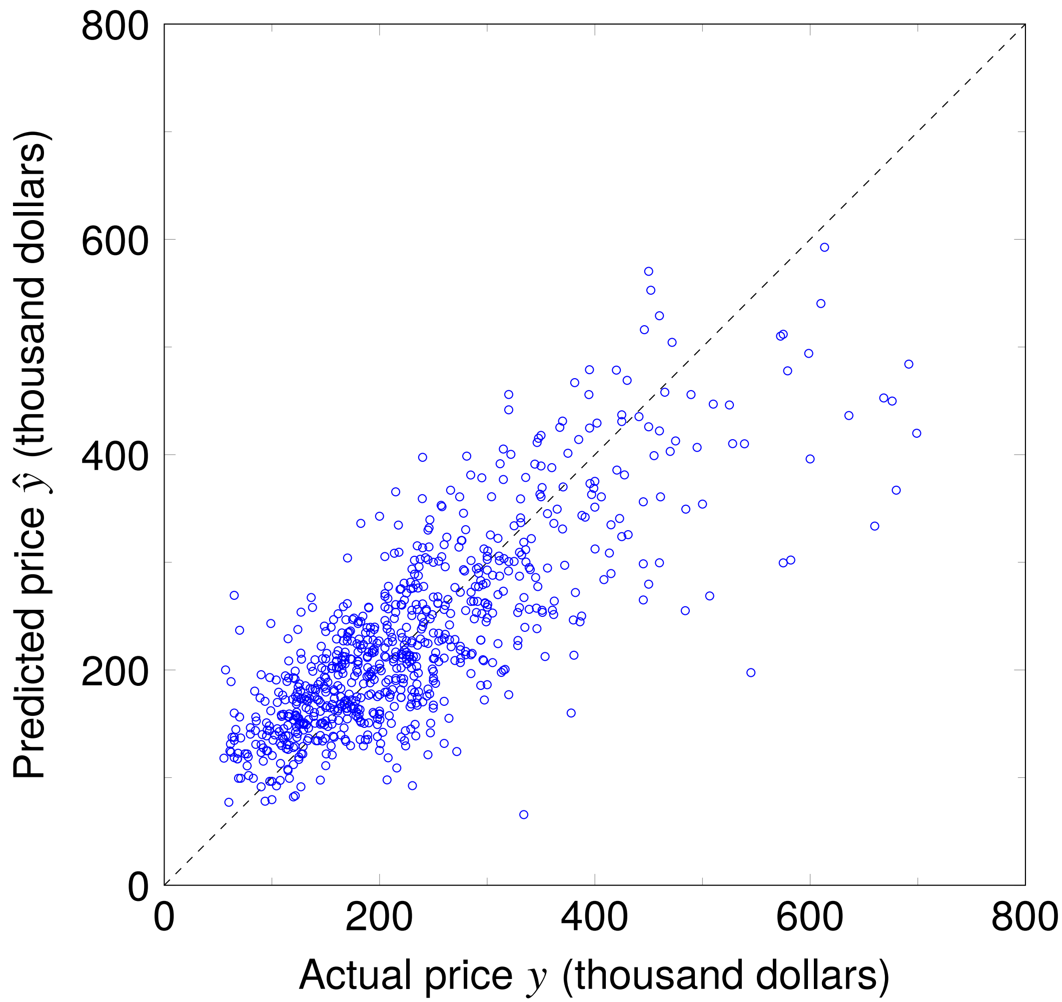
feature vector x has 7 elements

- x_1 is area of the house (in 1000 square feet)
- $x_2 = \max\{x_1 - 1.5, 0\}$, *i.e.*, area in excess of 1.5 (in 1000 square feet)
- x_3 is number of bedrooms
- x_4 is one for a condo; zero otherwise
- x_5, x_6, x_7 specify location (four groups of ZIP codes)

Location	x_5	x_6	x_7
A	0	0	0
B	1	0	0
C	0	1	0
D	0	0	1

Example: house price regression model

- use least squares to fit the eight model parameters v, β
- RMS fitting error is 68.3



Outline

- model fitting
- regression
- **linear-in-parameters models**
- time series examples
- validation
- least squares classification
- statistics interpretation

Linear-in-parameters model

we choose the model $\hat{f}(x)$ from a family of models

$$\hat{f}(x) = \theta_1 f_1(x) + \theta_2 f_2(x) + \cdots + \theta_p f_p(x)$$

- the functions f_i are scalar valued *basis functions* (chosen by us)
- the basis functions often include a constant function (typically, $f_1(x) = 1$)
- the coefficients $\theta_1, \ldots, \theta_p$ are the model *parameters*
- the model $\hat{f}(x)$ is linear in the parameters θ_i
- if $f_1(x) = 1$, this can be interpreted as a regression model

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

with parameters $v = \theta_1$, $\beta = \theta_{2:p}$ and new features $\tilde{x}$ generated from x:

$$\tilde{x}_1 = f_2(x), \quad \ldots, \quad \tilde{x}_p = f_p(x)$$

Least squares model fitting

- fit linear-in-parameters model to data set $(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})$
- residual for data sample i is

$$r^{(i)} = y^{(i)} - \hat{f}(x^{(i)}) = y^{(i)} - \theta_1 f_1(x^{(i)}) - \cdots - \theta_p f_p(x^{(i)})$$

- least squares model fitting: choose parameters θ by minimizing MSE

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

- this is a least squares problem: minimize $\|A\theta - y^{\mathrm{d}}\|^2$ with

$$A = \left[\begin{array}{ccc} f_1(x^{(1)}) & \cdots & f_p(x^{(1)}) \\ f_1(x^{(2)}) & \cdots & f_p(x^{(2)}) \\ \vdots & & \vdots \\ f_1(x^{(N)}) & \cdots & f_p(x^{(N)}) \end{array}\right], \qquad \theta = \left[\begin{array}{c} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{array}\right], \qquad y^{\mathrm{d}} = \left[\begin{array}{c} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{array}\right]$$

Example: polynomial approximation

$$\hat{f}(x) = \theta_1 + \theta_2 x + \theta_3 x^2 + \cdots + \theta_p x^{p-1}$$

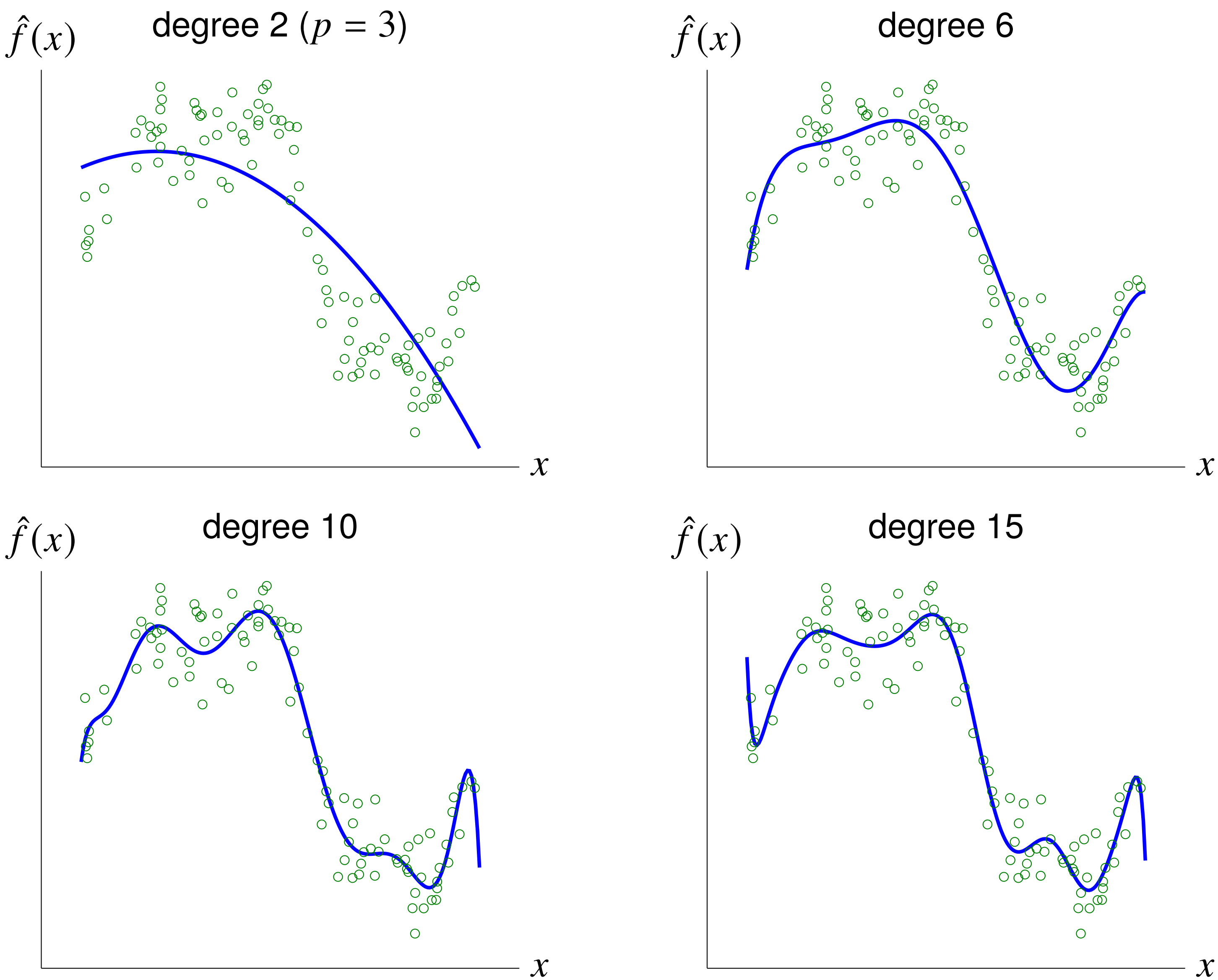
- a linear-in-parameters model with basis functions $1, x, \ldots, x^{p-1}$
- least squares model fitting: choose parameters θ by minimizing MSE

$$\frac{1}{N}\left((y^{(1)} - \hat{f}(x^{(1)}))^2 + (y^{(2)} - \hat{f}(x^{(2)}))^2 + \cdots + (y^{(N)} - \hat{f}(x^{(N)}))^2\right)$$

- in matrix notation: minimize $\|A\theta - y^{\mathrm{d}}\|^2$ with

$$A = \begin{bmatrix} 1 & x^{(1)} & (x^{(1)})^2 & \cdots & (x^{(1)})^{p-1} \\ 1 & x^{(2)} & (x^{(2)})^2 & \cdots & (x^{(2)})^{p-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x^{(N)} & (x^{(N)})^2 & \cdots & (x^{(N)})^{p-1} \end{bmatrix}, \qquad y^{\mathrm{d}} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

Example



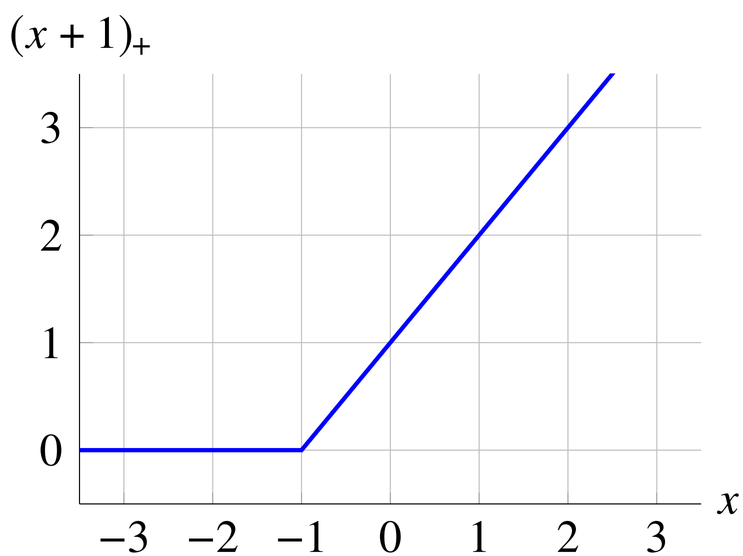
data set of 100 examples

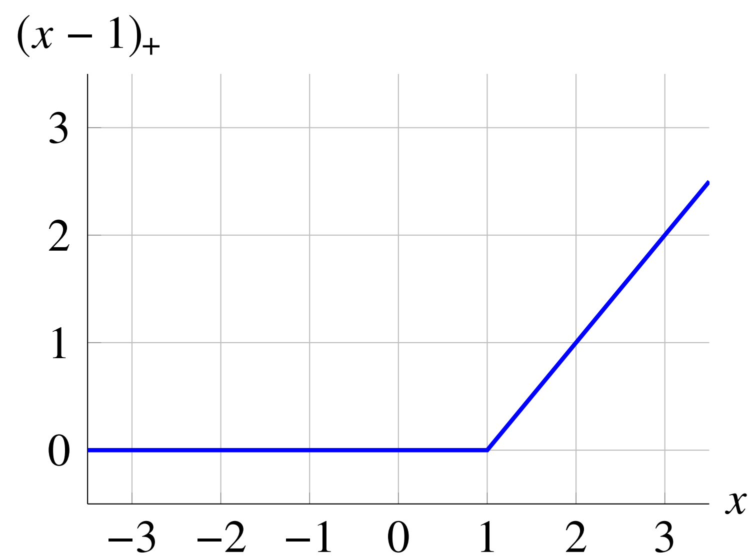
Piecewise-affine function

- define *knot points* $a_1 < a_2 < \cdots < a_k$ on the real axis
- piecewise-affine function is continuous, and affine on each interval $[a_k, a_{k+1}]$
- piecewise-affine function with knot points $a_1, \ldots, a_k$ can be written as

$$\hat{f}(x) = \theta_1 + \theta_2 x + \theta_3 (x - a_1)_+ + \cdots + \theta_{2+k}(x - a_k)_+$$

where $u_+ = \max\{u, 0\}$



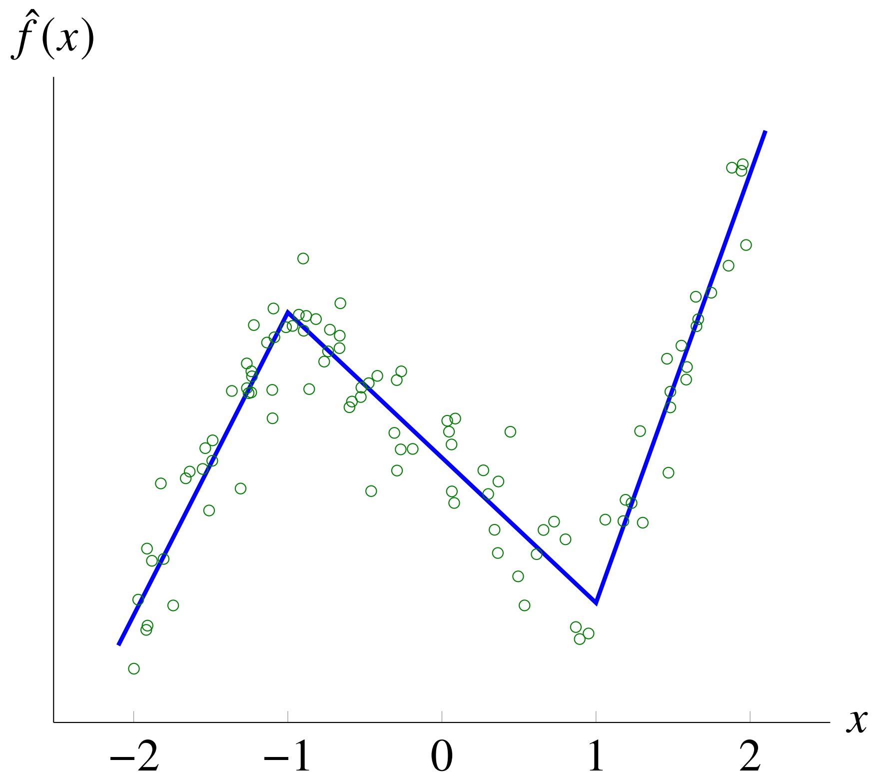


Piecewise-affine function fitting

piecewise-affine model is in linear in the parameters θ, with basis functions

$$f_1(x) = 1, \quad f_2(x) = x, \quad f_3(x) = (x - a_1)_+, \quad \ldots, \quad f_{k+2}(x) = (x - a_k)_+$$

Example: fit piecewise-affine function with knots $a_1 = -1$, $a_2 = 1$ to 100 points



Outline

- model fitting
- regression
- linear-in-parameters models
- **time series examples**
- validation
- least squares classification
- statistics interpretation

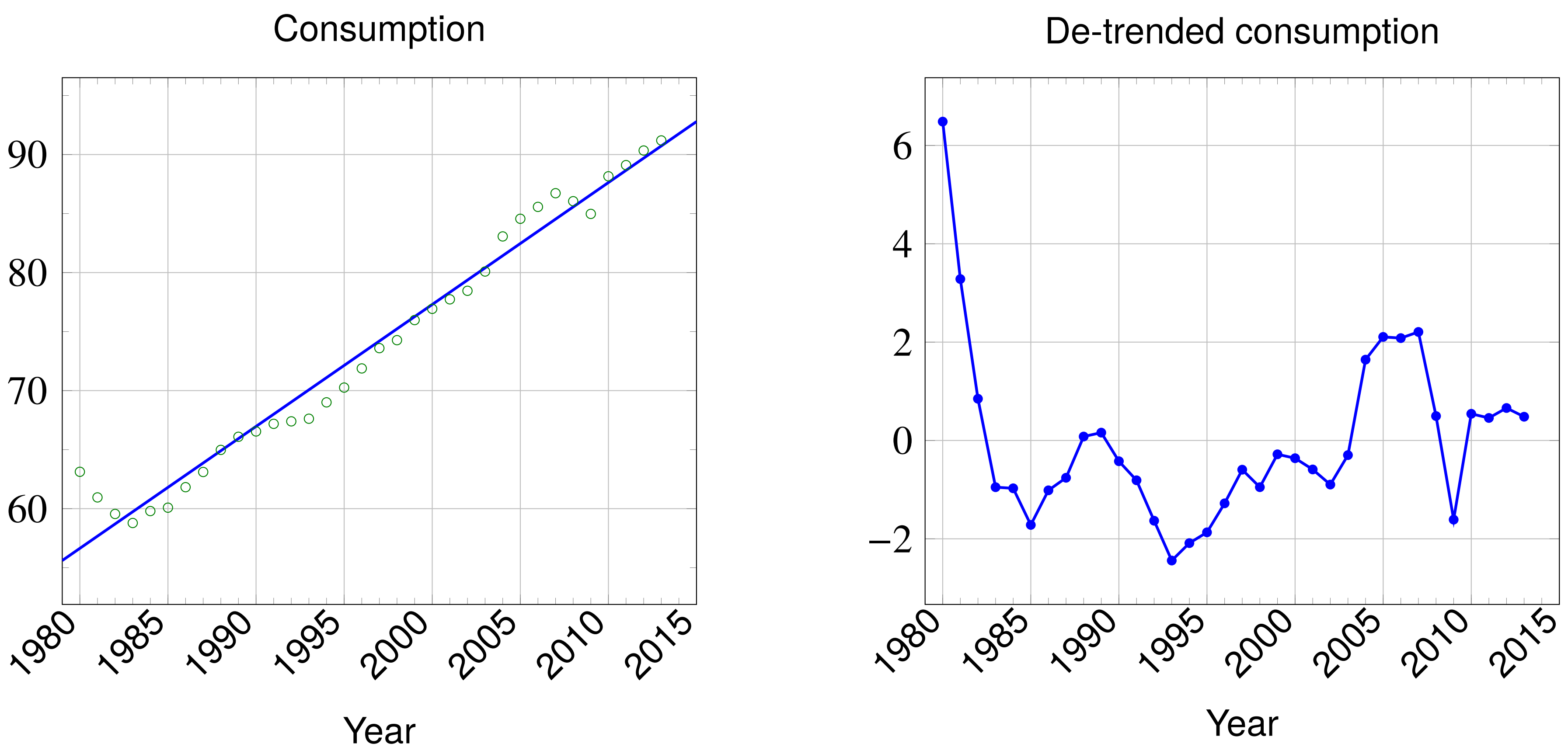
Time series trend

- N data samples from time series: $y^{(i)}$ is value at time i, for $i = 1, \ldots, N$
- straight-line fit $\hat{y}^{(i)} = \theta_1 + \theta_2 i$ is the *trend line*
- $y^{\mathrm{d}} - \hat{y}^{\mathrm{d}} = (y^{(1)} - \hat{y}^{(1)}, \ldots, y^{(N)} - \hat{y}^{(N)})$ is the *de-trended* time series
- least squares fitting of trend line: minimize $\|A\theta - y^{\mathrm{d}}\|^2$ with

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ \vdots & \vdots \\ 1 & N \end{bmatrix}, \qquad y^{\mathrm{d}} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

Example: world petroleum consumption

- time series of world petroleum consumption (million barrels/day) versus year
- left figure shows data samples and trend line
- right figure shows de-trended time series



Trend plus seasonal component

- model time series as a linear trend plus a periodic component with period P:

$$\hat{y}^{\mathrm{d}} = \hat{y}^{\mathrm{lin}} + \hat{y}^{\mathrm{seas}}$$

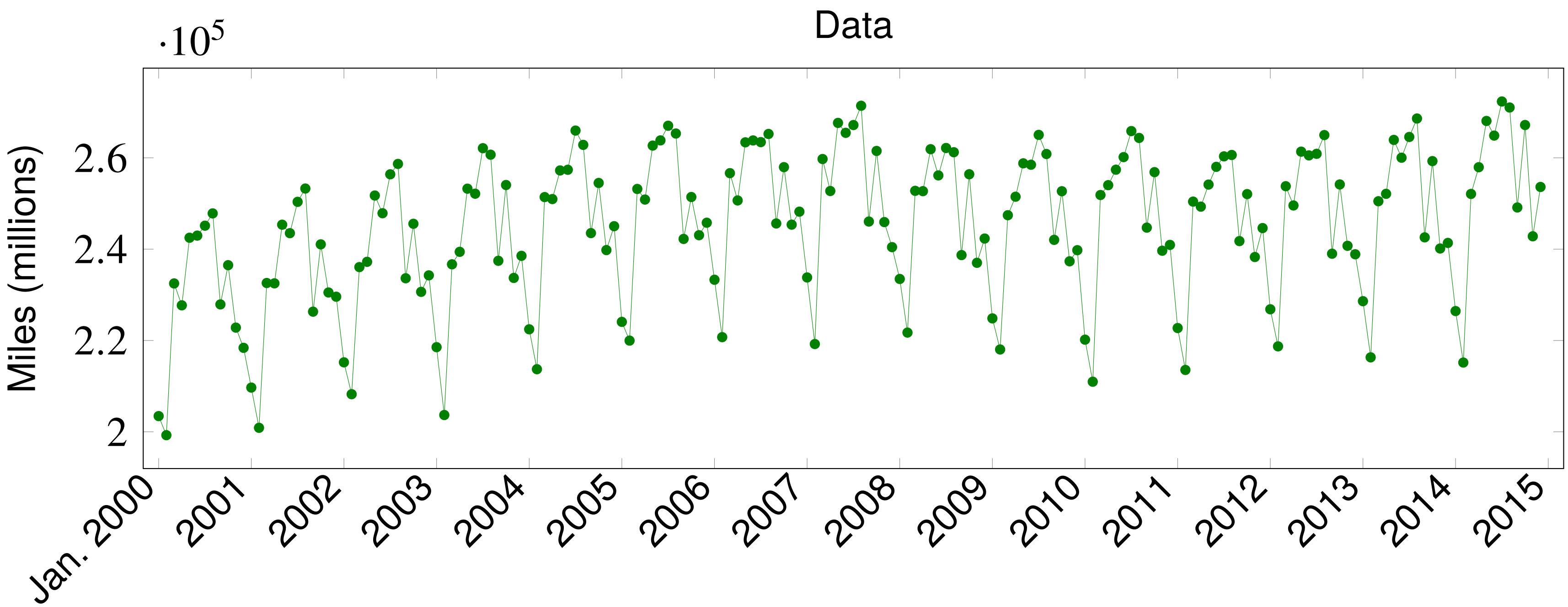
with $\hat{y}^{\mathrm{lin}} = \theta_1(1, 2, \ldots, N)$ and

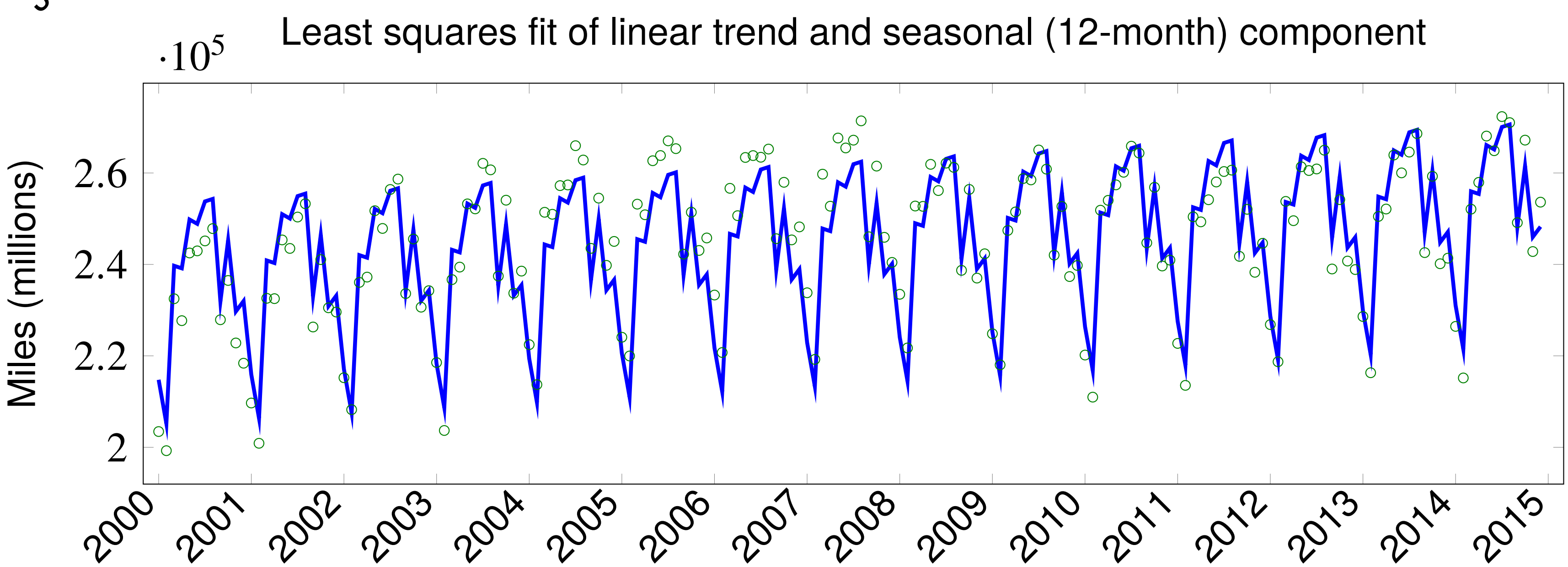
$$\hat{y}^{\mathrm{seas}} = (\theta_2, \theta_3, \ldots, \theta_{P+1}, \theta_2, \theta_3, \ldots, \theta_{P+1}, \ldots, \theta_2, \theta_3, \ldots, \theta_{P+1})$$

- the mean of $\hat{y}^{\mathrm{seas}}$ serves as a constant offset
- residual $y^{\mathrm{d}} - \hat{y}^{\mathrm{d}}$ is the *de-trended*, *seasonally adjusted* time series
- least squares formulation: minimize $\|A\theta - y^{\mathrm{d}}\|^2$ with

$$A_{1:N,1} = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ N \end{bmatrix}, \qquad A_{1:N,2:P+1} = \begin{bmatrix} I_P \\ I_P \\ \vdots \\ I_P \end{bmatrix}, \qquad y^{\mathrm{d}} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

Example: vehicle miles traveled in the US per month





Auto-regressive (AR) time series model

$$\hat{z}_{t+1} = \beta_1 z_t + \cdots + \beta_M z_{t-M+1}, \qquad t = M, M+1, \ldots$$

- $z_1, z_2, \ldots$ is a time series
- $\hat{z}_{t+1}$ is a prediction of z_{t+1}, made at time t
- prediction $\hat{z}_{t+1}$ is a linear function of previous M values $z_t, \ldots, z_{t-M+1}$
- M is the *memory* of the model

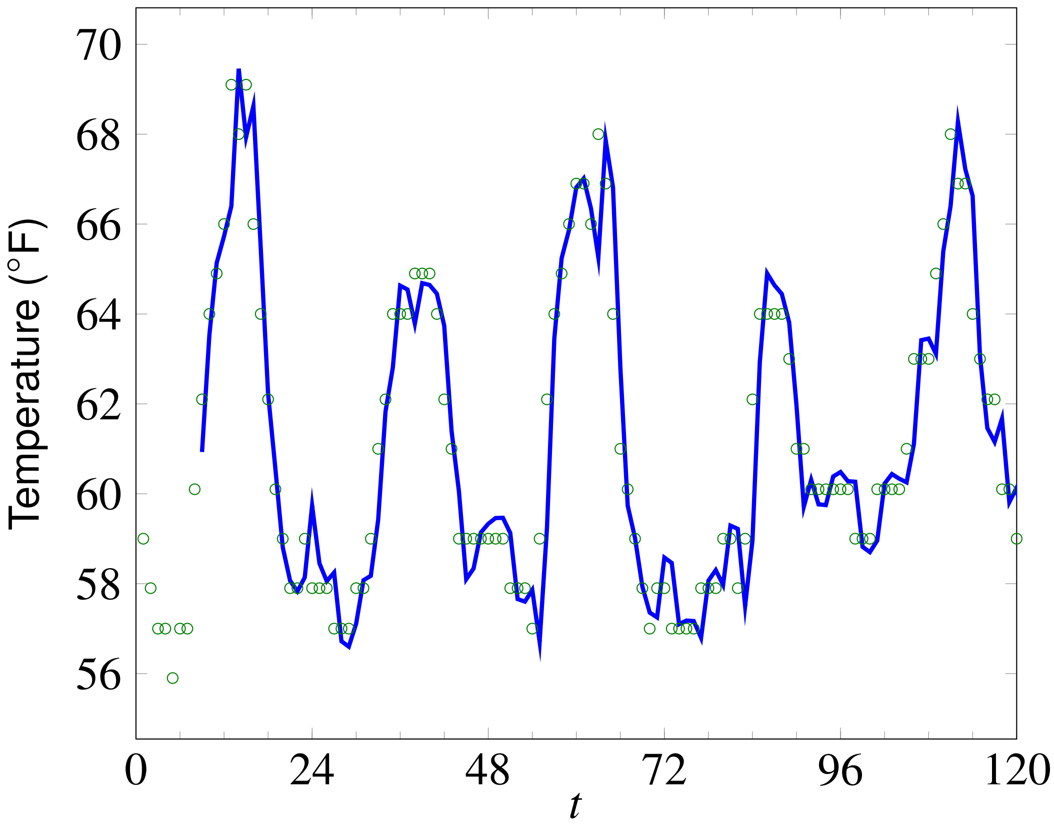
Least squares fitting of AR model: given oberved data $z_1, \ldots, z_T$, minimize

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

this is a least squares problem: minimize $\|A\beta - y^{\mathrm{d}}\|^2$ with

$$A = \begin{bmatrix} z_M & z_{M-1} & \cdots & z_1 \\ z_{M+1} & z_M & \cdots & z_2 \\ \vdots & \vdots & & \vdots \\ z_{T-1} & z_{T-2} & \cdots & z_{T-M} \end{bmatrix}, \qquad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_M \end{bmatrix}, \qquad y^{\mathrm{d}} = \begin{bmatrix} z_{M+1} \\ z_{M+2} \\ \vdots \\ z_T \end{bmatrix}$$

Example: hourly temperature at LAX



- blue line shows prediction by AR model of memory $M = 8$
- model was fit on time series of length $T = 744$ (May 1–31, 2016)
- plot shows first five days

Outline

- model fitting
- regression
- linear-in-parameters models
- time series examples
- **validation**
- least squares classification
- statistics interpretation

Generalization and validation

Generalization ability: ability of model to predict outcomes for new, unseen data

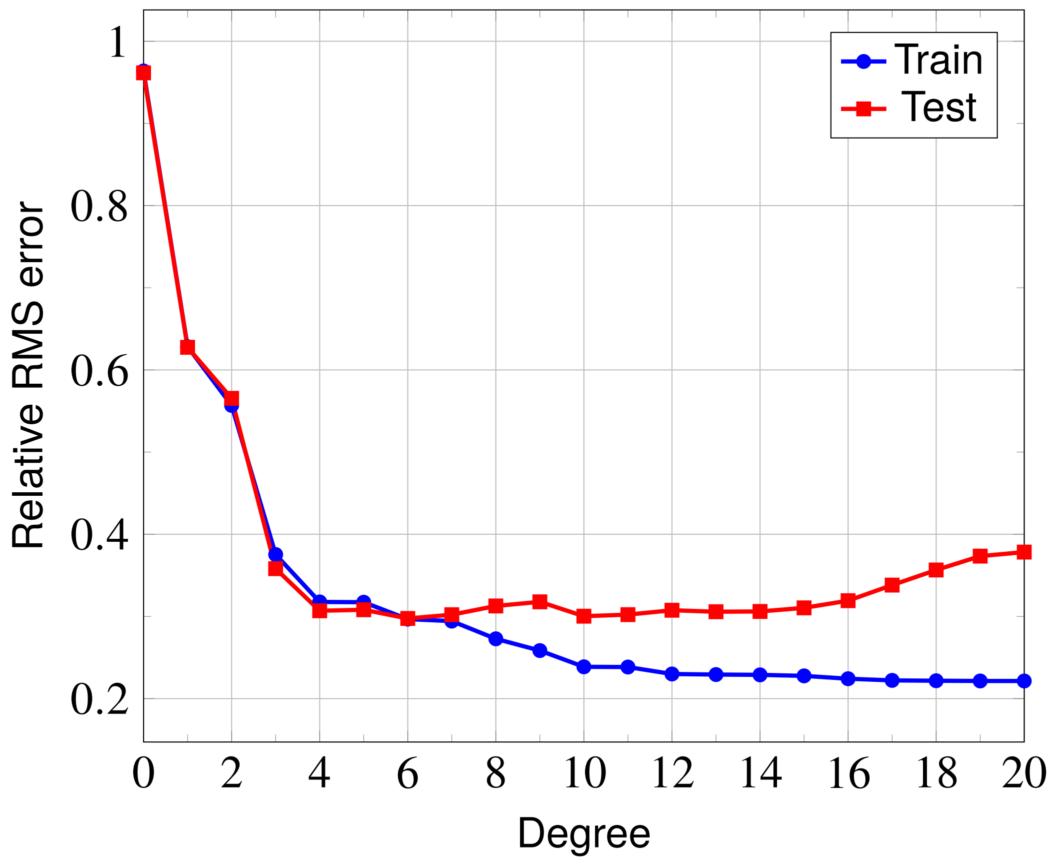
Model validation: to assess generalization ability,

- divide data in two sets: *training set* and *test (or validation)* set
- use training set to fit model
- use test set to get an idea of generalization ability
- this is also called *out-of-sample validation*

Over-fit model

- model with low prediction error on training set, bad generalization ability
- prediction error on training set is much smaller than on test set

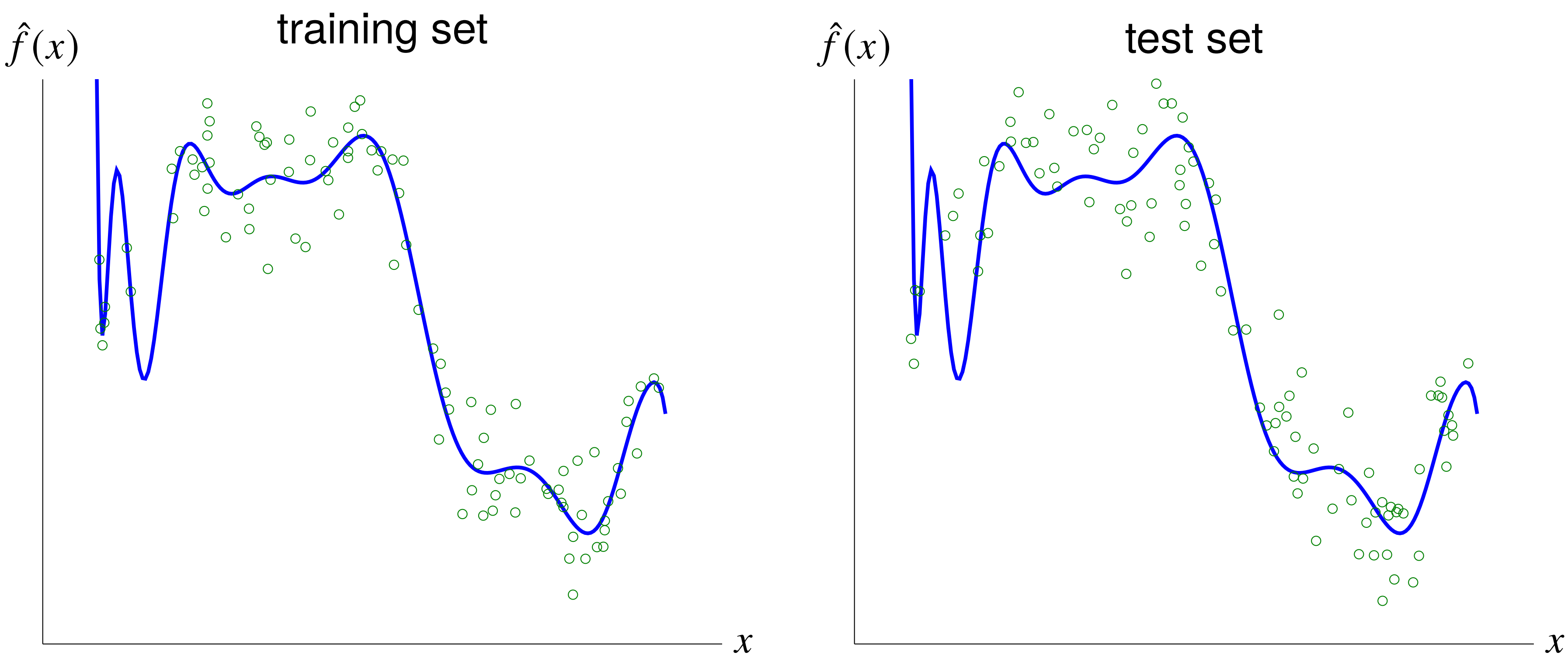
Example: polynomial fitting



- training set is data set of 100 points used on page 9.11
- test set is a similar set of 100 points
- plot suggests using degree 6

Over-fitting

polynomial of degree 20 on training and test set



over-fitting is evident at the left end of the interval

Cross-validation

an extension of out-of-sample validation

- divide data in K sets (*folds*); typical values are $K = 5$, $K = 10$
- for $i = 1$ to K, fit model i using fold i as test set and other data as training set
- compare parameters and train/test RMS errors for the K models

House price model (page 9.7) with 5 folds (155 or 154 examples each)

	Model parameters								RMS error	
Fold	v	β_1	β_2	β_3	β_4	β_5	β_6	β_7	Train	Test
1	122.5	166.9	−39.3	−16.3	−24.0	−100.4	−106.7	−26.0	67.3	72.8
2	101.0	186.7	−55.8	−18.7	−14.8	−99.1	−109.6	−17.9	67.8	70.8
3	133.6	167.2	−23.6	−18.7	−14.7	−109.3	−114.4	−28.5	69.7	63.8
4	108.4	171.2	−41.3	−15.4	−17.7	−94.2	−103.6	−29.8	65.6	78.9
5	114.5	185.7	−52.7	−20.9	−23.3	−102.8	−110.5	−23.4	70.7	58.3

Outline

- model fitting
- regression
- linear-in-parameters models
- time series example
- validation
- **least squares classification**
- statistics interpretation

Boolean (two-way) classification

- a data fitting problem where the outcome y can take two values $+1$, -1
- values of y represent two categories (true/false, spam/not spam, ...)
- model $\hat{y} = \hat{f}(x)$ is called a *Boolean classifier*

Least squares classifier

- use least squares to fit model $\tilde{f}(x)$ to training set $(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})$
- $\tilde{f}(x)$ can be a regression model $\tilde{f}(x) = x^T\beta + v$ or linear in parameters

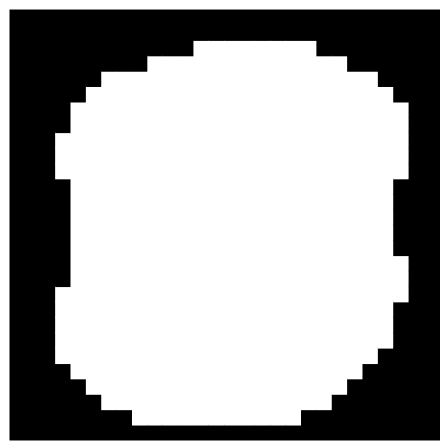
$$\tilde{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

- take sign of $\tilde{f}(x)$ to get a Boolean classifier

$$\hat{f}(x) = \mathrm{sign}(\tilde{f}(x)) = \begin{cases} +1 & \text{if } \tilde{f}(x) \geq 0 \\ -1 & \text{if } \tilde{f}(x) < 0 \end{cases}$$

Example: handwritten digit classification

- MNIST data set used in homework
- 28×28 images of handwritten digits ($n = 28^2 = 784$ pixels)
- data set contains 60000 training examples; 10000 test examples
- we only use the 493 pixels that are nonzero in at least 600 training examples

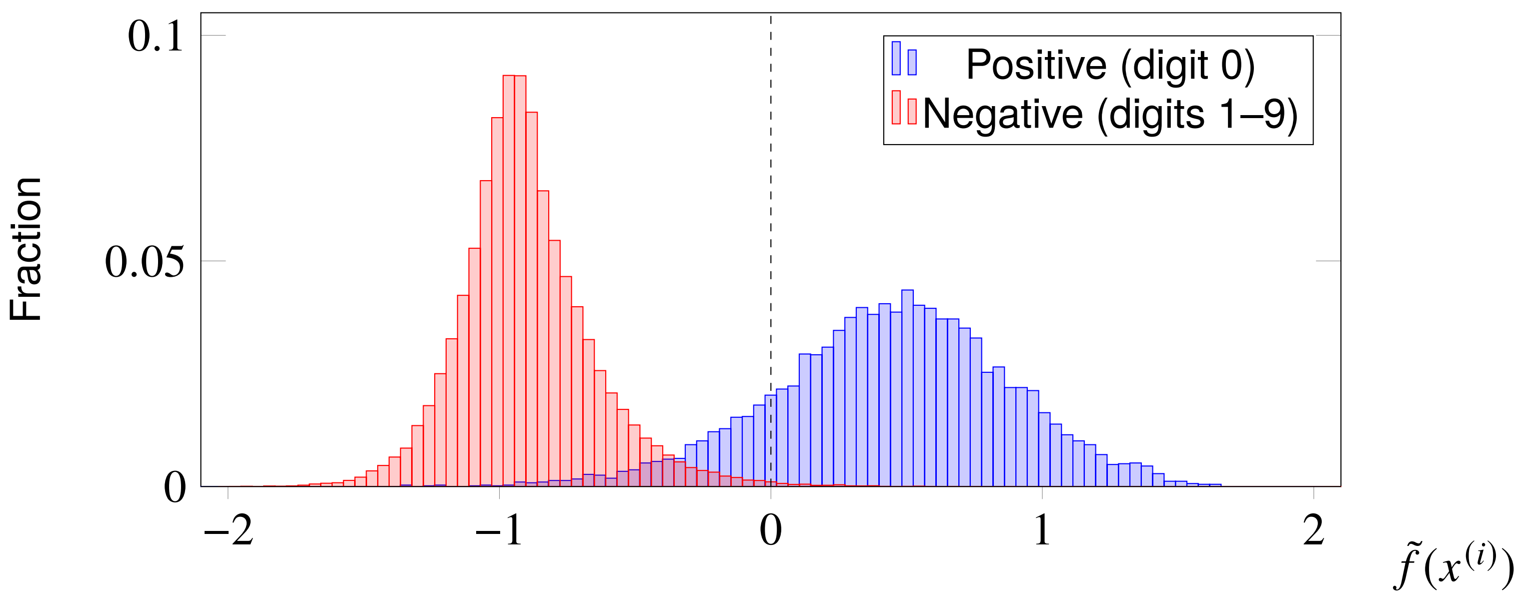


- Boolean classifier distinguishes digit zero ($y = 1$) from other digits ($y = -1$)

Classifier with basic regression model

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

- x is vector of 493 pixel intensities
- figure shows distribution of $\tilde{f}(x^{(i)}) = (x^{(i)})^T\hat{\beta} + \hat{v}$ on training set



- blue bars to the left of dashed line are false negatives (misclassified digits zero)
- red bars to the right of dashed line are false positives (misclassified non-zeros)

Prediction error

- for each data point x, y we have four combinations of prediction and outcome

	Prediction	
Outcome	$\hat{y} = +1$	$\hat{y} = -1$
$y = +1$	true positive	false negative
$y = -1$	false positive	true negative

- classifier can be evaluated by counting data points for each combination

Training set

	Prediction		
Outcome	$\hat{y} = +1$	$\hat{y} = -1$	Total
$y = +1$	5158	765	5923
$y = -1$	169	53910	54077
All	5325	54675	60000

error rate $(765 + 169)/60000 = 1.6\%$

Test set

	Prediction		
Outcome	$\hat{y} = +1$	$\hat{y} = -1$	Total
$y = +1$	864	116	980
$y = -1$	42	8978	9020
All	906	9094	10000

error rate $(116 + 42)/10000 = 1.6\%$

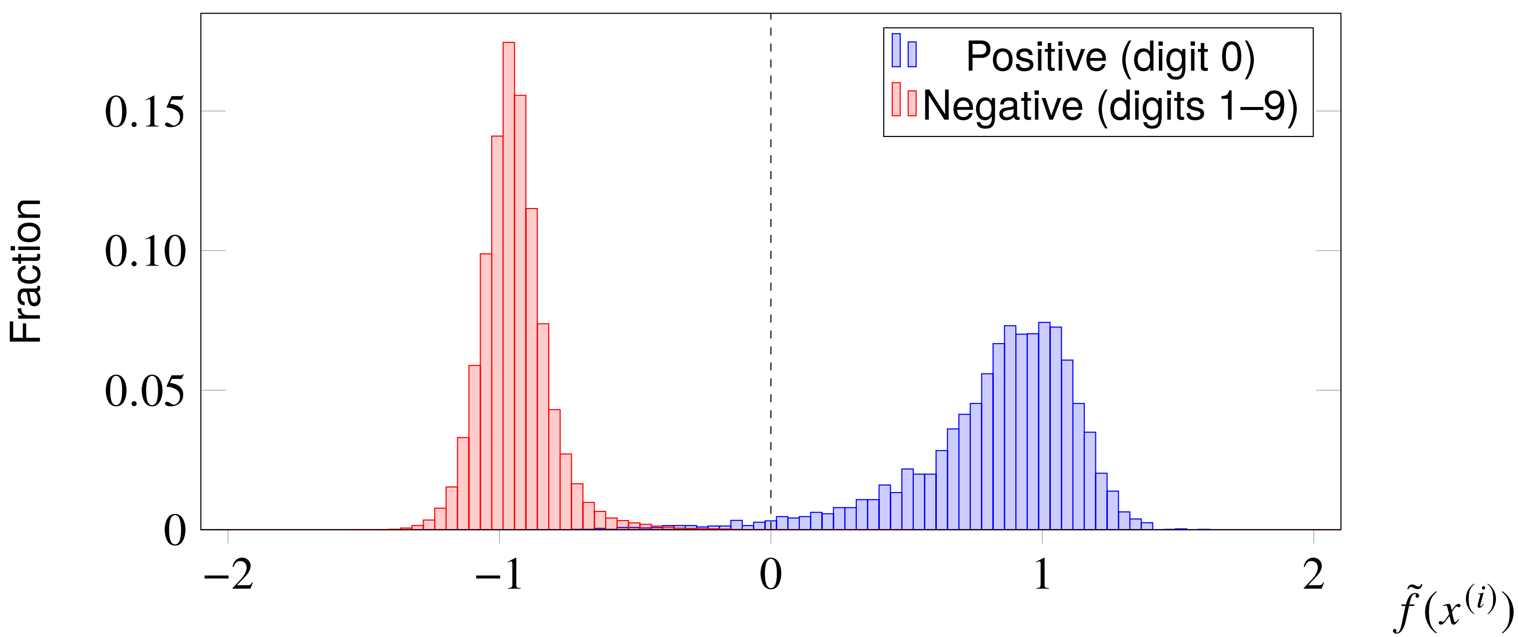
Classifier with additional nonlinear features

$$\hat{f}(x) = \mathrm{sign}(\tilde{f}(x)) = \mathrm{sign}(\sum_{i=1}^{p} \theta_i f_i(x))$$

- basis functions include constant, 493 elements of x, plus 5000 functions

$$f_i(x) = \max\{0, r_i^T x + s_i\} \qquad \text{with randomly generated } r_i,\ s_i$$

- figure shows distribution of $\tilde{f}(x^{(i)})$ on training set



Prediction error

Training set: error rate 0.21%

	Prediction		
Outcome	$\hat{y} = +1$	$\hat{y} = -1$	Total
$y = +1$	5813	110	5923
$y = -1$	15	54062	54077
All	5828	54172	60000

Test set: error rate 0.24%

	Prediction		
Outcome	$\hat{y} = +1$	$\hat{y} = -1$	Total
$y = +1$	963	17	980
$y = -1$	7	9013	9020
All	970	9030	10000

Multi-class classification

- a data fitting problem where the outcome y can takes values $1, \ldots, K$
- values of y represent K labels or categories
- multi-class classifier $\hat{y} = \hat{f}(x)$ maps x to an element of $\{1, 2, \ldots, K\}$

Least squares multi-class classifier

- for $k = 1, \ldots, K$, compute Boolean classifier to distinguish class k from not k

$$\hat{f}_k(x) = \mathrm{sign}(\tilde{f}_k(x))$$

- define multi-class classifier as

$$\hat{f}(x) = \underset{k=1,\ldots,K}{\operatorname{argmax}} \tilde{f}_k(x)$$

Example: handwritten digit classification

- we compute a least squares Boolean classifier for each digit versus the rest

$$\hat{f}_k(x) = \mathrm{sign}(x^T \beta_k + v_k), \quad k = 1, \ldots, K$$

- table shows results for test set (error rate 13.9%)

	Prediction										
Digit	0	1	2	3	4	5	6	7	8	9	Total
0	944	0	1	2	2	8	13	2	7	1	980
1	0	1107	2	2	3	1	5	1	14	0	1135
2	18	54	815	26	16	0	38	22	39	4	1032
3	4	18	22	884	5	16	10	22	20	9	1010
4	0	22	6	0	883	3	9	1	12	46	982
5	24	19	3	74	24	656	24	13	38	17	892
6	17	9	10	0	22	17	876	0	7	0	958
7	5	43	14	6	25	1	1	883	1	49	1028
8	14	48	11	31	26	40	17	13	756	18	974
9	16	10	3	17	80	0	1	75	4	803	1009
All	1042	1330	887	1042	1086	742	994	1032	898	947	10000

Example: handwritten digit classification

- ten least squares Boolean classifiers use 5000 new features (page 9.29)
- table shows results for test set (error rate 2.6%)

	Prediction										
Digit	0	1	2	3	4	5	6	7	8	9	Total
0	972	0	0	2	0	1	1	1	3	0	980
1	0	1126	3	1	1	0	3	0	1	0	1135
2	6	0	998	3	2	0	4	7	11	1	1032
3	0	0	3	977	0	13	0	5	8	4	1010
4	2	1	3	0	953	0	6	3	1	13	982
5	2	0	1	5	0	875	5	0	3	1	892
6	8	3	0	0	4	6	933	0	4	0	958
7	0	8	12	0	2	0	1	992	3	10	1028
8	3	1	3	6	4	3	2	2	946	4	974
9	4	3	1	12	11	7	1	3	3	964	1009
All	997	1142	1024	1006	977	905	956	1013	983	997	10000

Outline

- model fitting
- regression
- linear-in-parameters models
- time series example
- validation
- least squares classification
- **statistics interpretation**

Linear regression model

$$y = X\beta + \epsilon$$

- β is (non-random) p-vector of unknown *parameters*
- X is $n \times p$ (data matrix or *design matrix*, *i.e.*, result of experiment design)
- if there is an offset v, we include it in β and add a column of ones in X
- ϵ is a random n-vector (*random error* or *disturbance*)
- y is an observable random n-vector

- this notation differs from previous sections but is common in statistics
- we discuss methods for estimating parameters β from observations of y

Assumptions

- X is tall ($n > p$) with linearly independent columns
- random disturbances ϵ_i have zero mean

$$\mathbf{E}\,\epsilon_i = 0 \qquad \text{for } i = 1, \ldots, n$$

- random disturbances have equal variances σ^2

$$\mathbf{E}\,\epsilon_i^2 = \sigma^2 \qquad \text{for } i = 1, \ldots, n$$

- random disturbances are uncorrelated (have zero covariances)

$$\mathbf{E}\,(\epsilon_i \epsilon_j) = 0 \qquad \text{for } i, j = 1, \ldots, n \text{ and } i \neq j$$

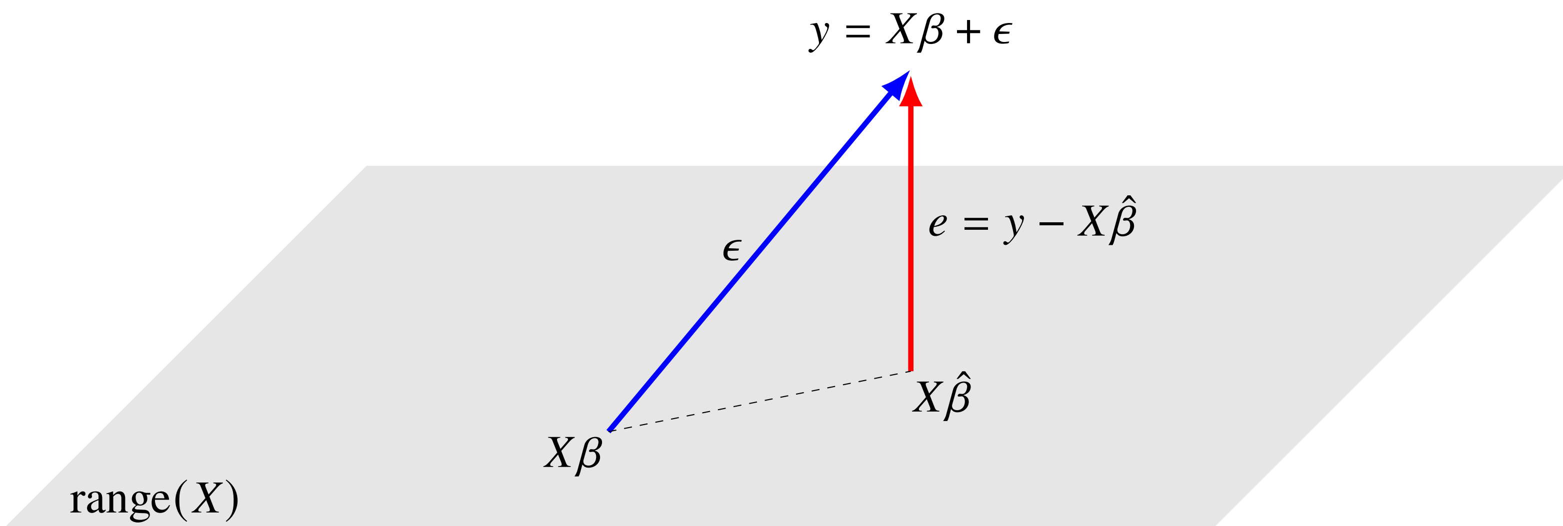
last three assumptions can be combined using matrix and vector notation:

$$\mathbf{E}\,\epsilon = 0, \qquad \mathbf{E}\,\epsilon\epsilon^T = \sigma^2 I$$

Least squares estimator

least squares estimate $\hat{\beta}$ of parameters β, given the observations y, is

$$\hat{\beta} = X^{\dagger} y = (X^T X)^{-1} X^T y$$



- $X\hat{\beta}$ is the orthogonal projection of y on $\mathrm{range}(X)$
- residual $e = y - X\hat{\beta}$ is an (observable) random variable

Mean and covariance of least squares estimate

$$\hat{\beta} = X^\dagger(X\beta + \epsilon) = \beta + X^\dagger \epsilon$$

- least squares estimator is *unbiased*: $\mathbf{E}\,\hat{\beta} = \beta$
- covariance matrix of least squares estimate is

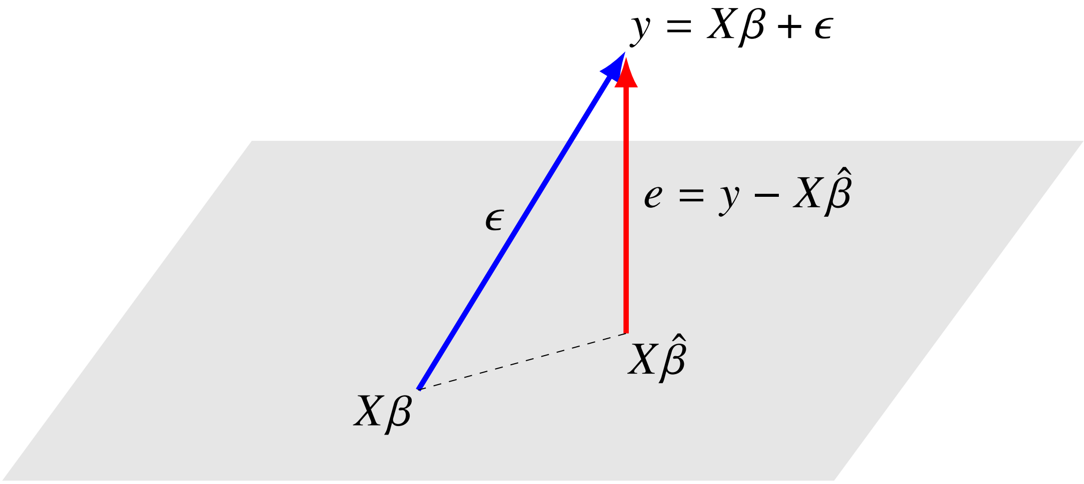
$$\begin{aligned}
\mathbf{E}\,(\hat{\beta} - \beta)(\hat{\beta} - \beta)^T &= \mathbf{E}\left((X^\dagger \epsilon)(X^\dagger \epsilon)^T\right) \\
&= \mathbf{E}\left((X^T X)^{-1} X^T \epsilon \epsilon^T X (X^T X)^{-1}\right) \\
&= \sigma^2 (X^T X)^{-1}
\end{aligned}$$

- covariance of $\hat{\beta}_i$ and $\hat{\beta}_j$ $(i \neq j)$ is

$$\mathbf{E}\left((\hat{\beta}_i - \beta_i)(\hat{\beta}_j - \beta_j)\right) = \sigma^2 \left((X^T X)^{-1}\right)_{ij}$$

for $i = j$, this is the variance of $\hat{\beta}_i$

Estimate of σ^2



$$\mathbf{E}\,\|\epsilon\|^2 = n\sigma^2$$

$$\mathbf{E}\,\|e\|^2 = (n-p)\sigma^2$$

$$\mathbf{E}\,\|X(\hat{\beta}-\beta)\|^2 = p\sigma^2$$

(proof on next page)

- define estimate $\hat{\sigma}$ of σ as

$$\hat{\sigma} = \frac{\|e\|}{\sqrt{n-p}}$$

- $\hat{\sigma}^2$ is an unbiased estimate of σ^2:

$$\mathbf{E}\,\hat{\sigma}^2 = \frac{1}{n-p}\,\mathbf{E}\,\|e\|^2 = \sigma^2$$

Proof.

first expression is immediate: $\mathbf{E}\,\|\epsilon\|^2 = \sum_{i=1}^{n} \mathbf{E}\,\epsilon_i^2 = n\sigma^2$

- to show that $\mathbf{E}\,\|X(\hat{\beta}-\beta)\|^2 = p\sigma^2$, first note that

$$\begin{aligned} X(\hat{\beta}-\beta) &= XX^\dagger y - X\beta \\ &= XX^\dagger(X\beta+\epsilon) - X\beta \\ &= XX^\dagger\epsilon \\ &= X(X^TX)^{-1}X^T\epsilon \end{aligned}$$

  on line 3 we used $X^\dagger X = I$ (however, note that $XX^\dagger \neq I$ if X is tall)

- squared norm of $X(\beta-\hat{\beta})$ is

$$\|X(\hat{\beta}-\beta)\|^2 = \epsilon^T(XX^\dagger)^2\epsilon = \epsilon^T XX^\dagger\epsilon$$

  first step uses symmetry of $XX^\dagger$; second step, $X^\dagger X = I$

- expected value of squared norm is

$$\begin{aligned}
\mathbf{E}\,\|X(\hat\beta-\beta)\|^2 = \mathbf{E}\left(\epsilon^T X X^\dagger \epsilon\right) &= \sum_{i,j} \mathbf{E}(\epsilon_i\epsilon_j)(XX^\dagger)_{ij} \\
&= \sigma^2 \sum_{i=1}^{n} (XX^\dagger)_{ii} \\
&= \sigma^2 \sum_{i=1}^{n}\sum_{j=1}^{p} X_{ij}(X^\dagger)_{ji} \\
&= \sigma^2 \sum_{j=1}^{p} (X^\dagger X)_{jj} \\
&= p\sigma^2
\end{aligned}$$

- expression $\mathbf{E}\,\|e\|^2 = (n-p)\sigma^2$ on page 9.38 now follows from

$$\|\epsilon\|^2 = \|e + X\hat\beta - X\beta\|^2 = \|e\|^2 + \|X(\hat\beta-\beta)\|^2$$

Linear estimator

linear regression model (page 9.34), with same assumptions as before (p. 9.35):

$$y = X\beta + \epsilon$$

a *linear estimator* of β maps observations y to the estimate

$$\hat{\beta} = By$$

- estimator is defined by the $p \times n$ matrix B
- least squares estimator is an example with $B = X^{\dagger}$

Unbiased linear estimator

if B is a left inverse of X, then estimator $\hat{\beta} = By$ can be written as:

$$\hat{\beta} = By = B(X\beta + \epsilon) = \beta + B\epsilon$$

- this shows that the linear estimator is *unbiased* ($\mathbf{E}\,\hat{\beta} = \beta$) if $BX = I$
- covariance matrix of unbiased linear estimator is

$$\mathbf{E}\left((\hat{\beta} - \beta)(\hat{\beta} - \beta)^T\right) = \mathbf{E}\left(B\epsilon\epsilon^T B^T\right) = \sigma^2 BB^T$$

- if c is a (non-random) p-vector, then estimate $c^T\hat{\beta}$ of $c^T\beta$ has variance

$$\mathbf{E}\,(c^T\hat{\beta} - c^T\beta)^2 = \sigma^2 c^T BB^T c$$

least squares estimator is an example with $B = X^\dagger$ and $BB^T = (X^T X)^{-1}$

Best linear unbiased estimator

if B is a left inverse of X then for all p-vectors c

$$c^T B B^T c \geq c^T (X^T X)^{-1} c$$

(proof on next page)

- left-hand side gives variance of $c^T \hat{\beta}$ for linear unbiased estimator

$$\hat{\beta} = By$$

- right-hand side gives variance of $c^T \hat{\beta}_{\mathrm{ls}}$ for least squares estimator

$$\hat{\beta}_{\mathrm{ls}} = X^{\dagger} y$$

- least squares estimator is the "*best linear unbiased estimator*" (BLUE)

this is known as the Gauss–Markov theorem

Proof.

- use $BX = I$ to write BB^T as

$$\begin{aligned} BB^T &= (B - (X^TX)^{-1}X^T)(B^T - X(X^TX)^{-1}) + (X^TX)^{-1} \\ &= (B - X^\dagger)(B - X^\dagger)^T + (X^TX)^{-1} \end{aligned}$$

- hence,

$$\begin{aligned} c^TBB^TC &= c^T(B - X^\dagger)(B - X^\dagger)^Tc + c^T(X^TX)^{-1}c \\ &= \|(B - X^\dagger)^Tc\|^2 + c^T(X^TX)^{-1}c \\ &\geq c^T(X^TX)^{-1}c \end{aligned}$$

  with equality if $B = X^\dagger$