9. Least squares data fitting

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

\[
x^{(1)}, \ldots, x^{(N)}, \quad y^{(1)}, \ldots, y^{(N)}
\]

- 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 \textit{regressors}
- the model is parameterized by the weight vector \( \beta \) and the offset (intercept) \( v \)
- the prediction error for example \( i \) is

\[ r^{(i)} = y^{(i)} - \hat{f}(x^{(i)}) \]
\[ = y^{(i)} - (x^{(i)})^T \beta - v \]

- 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 $v, \beta$ 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 $\|A\theta - y^d\|^2$ with

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

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)

<table>
<thead>
<tr>
<th>Location</th>
<th>( x_5 )</th>
<th>( x_6 )</th>
<th>( x_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Example: house price regression model

- use least squares to fit the eight model parameters $v, \beta$
- 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 \( \theta_i \)
• 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 \( x \):

\[
\tilde{x}_1 = f_2(x), \ldots, \tilde{x}_p = f_p(x)
\]
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 \(\theta\) 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 \(\|A\theta - y^d\|^2\) with
  \[
  A = \begin{bmatrix}
  f_1(x^{(1)}) & \cdots & f_p(x^{(1)}) \\
  f_1(x^{(2)}) & \cdots & f_p(x^{(2)}) \\
  \vdots & \ddots & \vdots \\
  f_1(x^{(N)}) & \cdots & f_p(x^{(N)})
  \end{bmatrix}, \quad \theta = \begin{bmatrix}
  \theta_1 \\
  \theta_2 \\
  \vdots \\
  \theta_p
  \end{bmatrix}, \quad y^d = \begin{bmatrix}
  y^{(1)} \\
  y^{(2)} \\
  \vdots \\
  y^{(N)}
  \end{bmatrix}
  \]
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 \(\theta\) 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 \(\|A\theta - y^d\|^2\) with

\[
A = \begin{bmatrix}
1 & x^{(1)} & (x^{(1)})^2 & \cdots & (x^{(1)})^{p-1} \\
1 & x^{(2)} & (x^{(2)})^2 & \cdots & (x^{(2)})^{p-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x^{(N)} & (x^{(N)})^2 & \cdots & (x^{(N)})^{p-1}
\end{bmatrix}, \quad y^d = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}
\]
Example

\[ \hat{f}(x) \quad \text{degree 2} \quad (p = 3) \]

\[ \hat{f}(x) \quad \text{degree 6} \]

\[ \hat{f}(x) \quad \text{degree 10} \]

\[ \hat{f}(x) \quad \text{degree 15} \]

data set of 100 examples

Least squares data fitting
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 $u_+ = \max \{ u, 0 \}$

Least squares data fitting 9.13
Piecewise-affine function fitting

piecewise-affine model is in linear in the parameters \( \theta \), 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
- $y^d - \hat{y}^d = (y^{(1)} - \hat{y}^{(1)}, \ldots, y^{(N)} - \hat{y}^{(N)})$ is the de-trended time series
- least squares fitting of trend line: minimize $\|A\theta - y^d\|^2$ with

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

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

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

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

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

$$A_{1:N,1} = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ N \end{bmatrix}, \quad A_{1:N,2:P+1} = \begin{bmatrix} I_P \\ I_P \\ \vdots \\ I_P \end{bmatrix}, \quad y^d = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$
Example: vehicle miles traveled in the US per month

Least squares fit of linear trend and seasonal (12-month) component

Least squares data fitting 9.18
Auto-regressive (AR) time series model

\[ \hat{z}_{t+1} = \beta_1 z_t + \cdots + \beta_M z_{t-M+1}, \quad t = M, M + 1, \ldots \]

- \( z_1, z_2, \ldots \) 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 observed 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 \( ||A\beta - y^d||^2 \) with

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

Least squares data fitting 9.22
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)

<table>
<thead>
<tr>
<th>Fold</th>
<th>$\nu$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\beta_3$</th>
<th>$\beta_4$</th>
<th>$\beta_5$</th>
<th>$\beta_6$</th>
<th>$\beta_7$</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>122.5</td>
<td>166.9</td>
<td>-39.3</td>
<td>-16.3</td>
<td>-24.0</td>
<td>-100.4</td>
<td>-106.7</td>
<td>-26.0</td>
<td>67.3</td>
<td>72.8</td>
</tr>
<tr>
<td>2</td>
<td>101.0</td>
<td>186.7</td>
<td>-55.8</td>
<td>-18.7</td>
<td>-14.8</td>
<td>-99.1</td>
<td>-109.6</td>
<td>-17.9</td>
<td>67.8</td>
<td>70.8</td>
</tr>
<tr>
<td>3</td>
<td>133.6</td>
<td>167.2</td>
<td>-23.6</td>
<td>-18.7</td>
<td>-14.7</td>
<td>-109.3</td>
<td>-114.4</td>
<td>-28.5</td>
<td>69.7</td>
<td>63.8</td>
</tr>
<tr>
<td>4</td>
<td>108.4</td>
<td>171.2</td>
<td>-41.3</td>
<td>-15.4</td>
<td>-17.7</td>
<td>-94.2</td>
<td>-103.6</td>
<td>-29.8</td>
<td>65.6</td>
<td>78.9</td>
</tr>
<tr>
<td>5</td>
<td>114.5</td>
<td>185.7</td>
<td>-52.7</td>
<td>-20.9</td>
<td>-23.3</td>
<td>-102.8</td>
<td>-110.5</td>
<td>-23.4</td>
<td>70.7</td>
<td>58.3</td>
</tr>
</tbody>
</table>
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 $\tilde{f}(x) = x^T \beta + \nu$ 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
  \[
  \hat{f}(x) = \text{sign}(\tilde{f}(x)) = \begin{cases} 
  +1 & \text{if } \tilde{f}(x) \geq 0 \\
  -1 & \text{if } \tilde{f}(x) < 0 
  \end{cases}
  \]
Example: handwritten digit classification

- MNIST data set used in homework
- $28 \times 28$ 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 ($y = -1$)

![Example Image]
Classifier with basic regression model

\[ \hat{f}(x) = \text{sign}(\tilde{f}(x)) = \text{sign}(x^T \beta + \nu) \]

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

![Histogram of \( \tilde{f}(x^{(i)}) \) for digit 0 and digits 1–9]

- 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

<table>
<thead>
<tr>
<th>Outcome</th>
<th>$\hat{y} = +1$</th>
<th>$\hat{y} = -1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y = +1$</td>
<td>true positive</td>
<td>false negative</td>
</tr>
<tr>
<td>$y = -1$</td>
<td>false positive</td>
<td>true negative</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>$\hat{y} = +1$</th>
<th>$\hat{y} = -1$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y = +1$</td>
<td>5158</td>
<td>765</td>
<td>5923</td>
</tr>
<tr>
<td>$y = -1$</td>
<td>169</td>
<td>53910</td>
<td>54077</td>
</tr>
<tr>
<td>All</td>
<td>5325</td>
<td>54675</td>
<td>60000</td>
</tr>
</tbody>
</table>

Training set

Test set

<table>
<thead>
<tr>
<th></th>
<th>$\hat{y} = +1$</th>
<th>$\hat{y} = -1$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y = +1$</td>
<td>864</td>
<td>116</td>
<td>980</td>
</tr>
<tr>
<td>$y = -1$</td>
<td>42</td>
<td>8978</td>
<td>9020</td>
</tr>
<tr>
<td>All</td>
<td>906</td>
<td>9094</td>
<td>10000</td>
</tr>
</tbody>
</table>

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

error rate $(116 + 42)/10000 = 1.6\%$
Classifier with additional nonlinear features

\[ \hat{f}(x) = \text{sign} (\tilde{f}(x)) = \text{sign} \left( \sum_{i=1}^{p} \theta_i f_i(x) \right) \]

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

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

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

## Training set: error rate 0.21%

<table>
<thead>
<tr>
<th>Outcome</th>
<th>( \hat{y} = +1 )</th>
<th>( \hat{y} = -1 )</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y = +1 )</td>
<td>5813</td>
<td>110</td>
<td>5923</td>
</tr>
<tr>
<td>( y = -1 )</td>
<td>15</td>
<td>54062</td>
<td>54077</td>
</tr>
<tr>
<td>All</td>
<td>5828</td>
<td>54172</td>
<td>60000</td>
</tr>
</tbody>
</table>

## Test set: error rate 0.24%

<table>
<thead>
<tr>
<th>Outcome</th>
<th>( \hat{y} = +1 )</th>
<th>( \hat{y} = -1 )</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y = +1 )</td>
<td>963</td>
<td>17</td>
<td>980</td>
</tr>
<tr>
<td>( y = -1 )</td>
<td>7</td>
<td>9013</td>
<td>9020</td>
</tr>
<tr>
<td>All</td>
<td>970</td>
<td>9030</td>
<td>10000</td>
</tr>
</tbody>
</table>

Least squares data fitting 9.30
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 $k$

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

• define multi-class classifier as

$$ \hat{f}(x) = \arg\max_{k=1,\ldots,K} \tilde{f}_k(x) $$
Example: handwritten digit classification

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

\[ f_k(x) = \text{sign}(x^T \beta_k + v_k), \quad k = 1, \ldots, K \]

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

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

Least squares data fitting 9.32
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%)

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

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

\[ y = X\beta + \epsilon \]

- \( \beta \) 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 \( \nu \), we include it in \( \beta \) and add a column of ones in \( X \)
- \( \epsilon \) 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 \( \beta \) from observations of \( y \)
Assumptions

• \( X \) is tall \((n > p)\) with linearly independent columns

• random disturbances \( \epsilon_i \) have zero mean

\[
E \epsilon_i = 0 \quad \text{for } i = 1, \ldots, n
\]

• random disturbances have equal variances \( \sigma^2 \)

\[
E \epsilon_i^2 = \sigma^2 \quad \text{for } i = 1, \ldots, n
\]

• random disturbances are uncorrelated (have zero covariances)

\[
E (\epsilon_i \epsilon_j) = 0 \quad \text{for } i, j = 1, \ldots, n \text{ and } i \neq j
\]

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

\[
E \epsilon = 0, \quad E \epsilon \epsilon^T = \sigma^2 I
\]
least squares estimate $\hat{\beta}$ of parameters $\beta$, given the observations $y$, is

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

$y = X\beta + \epsilon$

- $X\hat{\beta}$ is the orthogonal projection of $y$ on 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*: \( E \hat{\beta} = \beta \)
- covariance matrix of least squares estimate is

\[
E (\beta - \beta)(\hat{\beta} - \beta)^T = E \left( (X^\dagger \epsilon)(X^\dagger \epsilon)^T \right) = E \left( (X^T X)^{-1} X^T \epsilon \epsilon^T X (X^T X)^{-1} \right) = \sigma^2 (X^T X)^{-1}
\]

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

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

for \( i = j \), this is the variance of \( \hat{\beta}_i \)
Estimate of $\sigma^2$

- Define estimate $\hat{\sigma}$ of $\sigma$ as

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

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

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

$$E \|e\|^2 = n\sigma^2$$

$$E \|e\|^2 = (n - p)\sigma^2$$

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

(proof on next page)
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

\[
X(\hat{\beta} - \beta) = XX^\dagger y - X\beta = XX^\dagger (X\beta + \epsilon) - X\beta = XX^\dagger \epsilon = X(X^T X)^{-1} X^T \epsilon
\]

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

\[ E \|X(\hat{\beta} - \beta)\|^2 = E \left( \epsilon^T XX^\dagger \epsilon \right) = \sum_{i,j} E(\epsilon_i \epsilon_j) (XX^\dagger)_{ij} \]

\[ = \sigma^2 \sum_{i=1}^{n} (XX^\dagger)_{ii} \]

\[ = \sigma^2 \sum_{i=1}^{n} \sum_{j=1}^{p} X_{ij} (X^\dagger)_{ji} \]

\[ = \sigma^2 \sum_{j=1}^{p} (X^\dagger X)_{jj} \]

\[ = p\sigma^2 \]

• expression \( E \|e\|^2 = (n - p)\sigma^2 \) 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 \( \beta \) 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 ($E\hat{\beta} = \beta$) if $BX = I$

- covariance matrix of unbiased linear estimator is

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

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

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

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

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

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

(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 $c^T \hat{\beta}_{ls}$ for least squares estimator

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

- 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

\[
BB^T = (B - (XTX)^{-1}XT)(B^T - X(XTX)^{-1}) + (XTX)^{-1}
\]

\[
= (B - X^\dagger)(B - X^\dagger)^T + (XTX)^{-1}
\]

• hence,

\[
c^TBB^TC = c^T(B - X^\dagger)(B - X^\dagger)^Tc + c^T(XTX)^{-1}c
\]

\[
= \| (B - X^\dagger)^Tc \|^2 + c^T(XTX)^{-1}c
\]

\[
\geq c^T(XTX)^{-1}c
\]

with equality if $B = X^\dagger$