Computational Statistics

Summer 2007

Peter Bühlmann
with changes by Martin Mächler

Seminar für Statistik
ETH Zürich

March 2007 (March 21, 2007)

Contents

Chapter 1

Multiple Linear Regression

1.1 Introduction

Linear regression is a widely used statistical model in a broad variety of applications. It is one of the easiest examples to demonstrate important aspects of statistical modelling.

1.2 The Linear Model

Multiple Regression Model:
Given is one **response variable**: up to some random errors
it is a linear function of several **predictors** (or **covariables**).
The linear function involves unknown parameters. The goal is to estimate these parameters, to study their relevance and to estimate the error variance.

Model formula:

$$Y_i = \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \varepsilon_i \qquad (i = 1, \ldots, n) \tag{1.1}$$

Usually we assume that $\varepsilon_1, \ldots, \varepsilon_n$ are i.i.d. (**i**ndependent, **i**dentically **d**istributed) with $\mathbb{E}[\varepsilon_i] = 0,\ \mathrm{Var}(\varepsilon_i) = \sigma^2$.

Notations:

- $\mathbf{Y} = \{Y_i; i = 1, \ldots, n\}$ is the vector of the **response variables**
- $\mathbf{x}^{(j)} = \{x_{ij}; i = 1, \ldots, n\}$ is the vector of the jth predictor (covariable) $(j = 1, \ldots, p)$
- $\mathbf{x}_i = \{x_{ij}; j = 1, \ldots p\}$ is the vector of predictors for the ith observation $(i = 1, \ldots, n)$
- $\boldsymbol{\beta} = \{\beta_j; j = 1, \ldots, p\}$ is the vector of the unknown parameters
- $\boldsymbol{\varepsilon} = \{\varepsilon_i; i = 1, \ldots, n\}$ is the vector of the unknown random **errors**
- n is the sample size, p is the number of predictors

The parameters β_j and σ^2 are unknown and the errors ε_i are unobservable. On the other hand, the response variables Y_i and the predictors x_{ij} have been observed.

Model in vector notation:

$$\boxed{Y_i = \mathbf{x}_i^\intercal \boldsymbol{\beta} + \varepsilon_i \qquad (i = 1, \ldots, n)}$$

Model in matrix form:

$$\boxed{\begin{array}{ccccccc} \mathbf{Y} & = & X & \times & \boldsymbol{\beta} & + & \boldsymbol{\varepsilon} \\ n \times 1 & & n \times p & & p \times 1 & & n \times 1 \end{array}} \tag{1.2}$$

where X is a $(n \times p)$-matrix with rows $\mathbf{x}_i^\intercal$ and columns $\mathbf{x}^{(j)}$.

The first predictor variable is often a constant, i.e., $x_{i1} \equiv 1$ for all i. We then get an intercept in the model.

$$Y_i = \beta_1 + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \varepsilon_i.$$

We typically assume that the sample size n is larger than the number of predictors p, $n > p$, and moreover that the matrix X has full rank p, i.e., the p column vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(p)}$ are linearly independent.

1.2.1 Stochastic Models

The linear model in (1.1) involves some stochastic (random) components: the error terms ε_i are random variables and hence the response variables Y_i as well. The predictor variables x_{ij} are here assumed to be non-random. In some applications, however it is more appropriate to treat the predictor variables as random.

The stochastic nature of the error terms ε_i can be assigned to various sources: for example, measurement errors or inability to capture all underlying non-systematic effects which are then summarized by a random variable with expectation zero. The stochastic modelling approach will allow to quantify uncertainty, to assign significance to various components, e.g. significance of predictor variables in model (1.1), and to find a good compromise between the size of a model and the ability to describe the data (see section 1.7).

The observed response in the data is always assumed to be realizations of the random variables $Y_1, \ldots, Y_n$; the x_{ij}'s are non-random and equal to the observed predictors in the data.

1.2.2 Examples

Two-sample model:

$$p = 2, \qquad X = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}.$$

Main questions: $\mu_1 = \mu_2$? Quantitative difference between μ_1 and μ_2?
From introductory courses we know that one could use the two-sample t-test or two-sample Wilcoxon test.

Regression through the origin: $Y_i = \beta x_i + \varepsilon_i$ $(i = 1, \dots n)$.

$$p = 1, \qquad X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \qquad \boldsymbol{\beta} = \beta.$$

Simple linear regression: $Y_i = \beta_1 + \beta_2 x_i + \varepsilon_i$ $(i = 1, \dots n)$.

$$p = 2 \qquad X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}.$$

Quadratic regression: $Y_i = \beta_1 + \beta_2 x_i + \beta_3 x_i^2 + \varepsilon_i$ $(i = 1, \dots n)$.

$$p = 3, \qquad X = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}.$$

Note that the fitted function is quadratic in the x_i's but *linear* in the coefficients β_j and therefore a special case of the linear model (1.1).

Regression with transformed predictor variables:
$Y_i = \beta_1 + \beta_2 \log(x_{i2}) + \beta_3 \sin(\pi x_{i3}) + \varepsilon_i$ $(i = 1, \dots n)$.

$$p = 3, \qquad X = \begin{pmatrix} 1 & \log(x_{12}) & \sin(\pi x_{13}) \\ 1 & \log(x_{22}) & \sin(\pi x_{23}) \\ \vdots & \vdots & \vdots \\ 1 & \log(x_{n2}) & \sin(\pi x_{n3}) \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}.$$

Again, the model is *linear* in the coefficients β_j but nonlinear in the x_{ij}'s.

In summary:

> The model in (1.1) is called linear because it is linear in the coefficients β_j.
> The predictor (and also the response) variables can be transformed versions of the original predictor and/or response variables.

1.2.3 Goals of a linear regression analysis

- **A good "fit".** Fitting or estimating a (hyper-)plane over the predictor variables to explain the response variables such that the errors are "small". The standard tool for this is the method of *least squares* (see section 1.3).
- **Good parameter estimates.** This is useful to describe the change of the response when varying some predictor variable(s).
- **Good prediction.** This is useful to predict a new response as a function of new predictor variables.

- **Uncertainties and significance for the three goals above.** Confidence intervals and statistical tests are useful tools for this goal.
- **Development of a good model.** In an interactive process, using methods for the goals mentioned above, we may change parts of an initial model to come up with a better model.

The first and third goal can become conflicting, see section 1.7.

1.3 Least Squares Method

We assume the linear model $\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. We are looking for a "good" estimate of $\boldsymbol{\beta}$. The least squares estimator $\widehat{\boldsymbol{\beta}}$ is defined as

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{Y} - X\boldsymbol{\beta}\|^2 , \tag{1.3}$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$.

1.3.1 The normal equations

The minimizer in (1.3) can be computed explicitly (assuming that X has rank p). Computing partial derivatives of $\|\mathbf{Y} - X\boldsymbol{\beta}\|^2$ with respect to $\boldsymbol{\beta}$ (p-dimensional gradient vector), evaluated at $\widehat{\boldsymbol{\beta}}$, and setting them to zero yields

$$(-2)\ X^\intercal(\mathbf{Y} - X\widehat{\boldsymbol{\beta}}) = \mathbf{0} \qquad ((p \times 1) - \text{null-vector}).$$

Thus, we get the **normal equations**

$$X^\intercal X\widehat{\boldsymbol{\beta}} = X^\intercal \mathbf{Y}. \tag{1.4}$$

These are p linear equations for the p unknowns (components of $\widehat{\boldsymbol{\beta}}$).

Assuming that the matrix X has full rank p, the $p \times p$ matrix $X^\intercal X$ is invertible, the least squares estimator is unique and can be represented as

$$\boxed{\widehat{\boldsymbol{\beta}} = (X^\intercal X)^{-1} X^\intercal \mathbf{Y}.}$$

This formula is useful for theoretical purposes. For numerical computation it is much more stable to use the QR decomposition instead of inverting the matrix $X^\intercal X$. [1]

From the *residuals* $r_i = Y_i - \mathbf{x}_i\widehat{\boldsymbol{\beta}}$, the usual estimate for σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n} r_i^2.$$

Note that the r_i's are estimates for ε_i's; hence the estimator is plausible, up to the somewhat unusual factor $1/(n-p)$. It will be shown in section 1.4.1 that due to this factor, $\mathbb{E}[\hat{\sigma}^2] = \sigma^2$.

[1] Let $X = QR$ with orthogonal $(n \times p)$ matrix Q and upper (**R**ight) triangular $(p \times p)$ R. Because of $X^\intercal X = R^\intercal Q^\intercal QR = R^\intercal R$, computing $\boldsymbol{\beta}$ only needs subsequent solution of two triangular systems: First solve $R^\intercal \mathbf{c} = X^\intercal \mathbf{Y}$ for $\mathbf{c}$, and then solve $R\widehat{\boldsymbol{\beta}} = \mathbf{c}$.

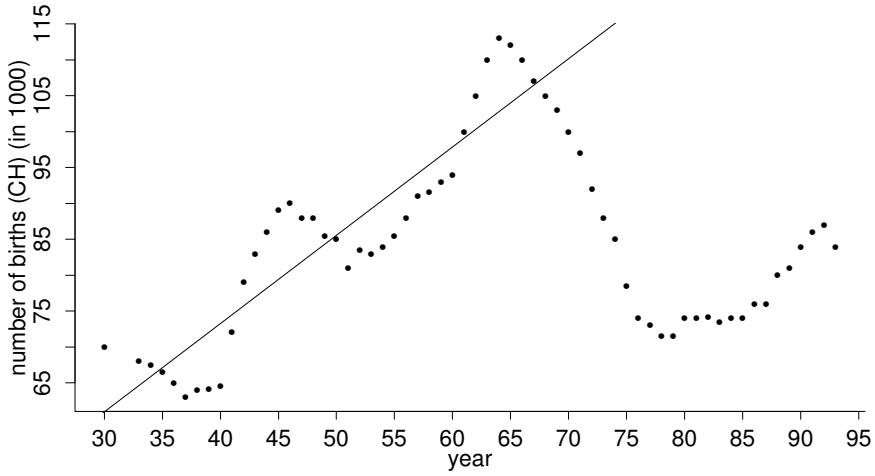


Figure 1.1: The pill kink.

1.3.2 Assumptions for the Linear Model

We emphasize here that we do not make any assumptions on the predictor variables, except that the matrix X has full rank $p < n$. In particular, the predictor variables can be continuous or discrete (e.g. binary).

We need some assumptions so that fitting a linear model by least squares is reasonable and that tests and confidence intervals (see 1.5) are approximately valid.

1. **The linear regression equation is correct.** This means: $\mathbb{E}[\varepsilon_i] = 0$ for all i.
2. **All $\mathbf{x}_i$'s are exact.** This means that we can observe them perfectly.
3. **The variance of the errors is constant ("homoscedasticity").** This means: $\mathrm{Var}(\varepsilon_i) = \sigma^2$ for all i.
4. **The errors are uncorrelated.** This means: $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$ for all $i \neq j$.
5. **The errors $\{\varepsilon_i;\ i = 1, \ldots, n\}$ are jointly normally distributed.** This implies that also $\{Y_i; i = 1, \ldots, n\}$ are jointly normally distributed.

In case of violations of item 3, we can use weighted least squares instead of least squares. Similarly, if item 4 is violated, we can use generalized least squares. If the normality assumption in 5 does not hold, we can use robust methods instead of least squares. If assumption 2 fails to be true, we need corrections known from "errors in variables" methods. If the crucial assumption in 1 fails, we need other models than the linear model.

The following example shows violations of assumption 1 and 4. The response variable is the annual number of births in Switzerland since 1930, and the predictor variable is the time (year).

We see in Figure 1.1 that the data can be approximately described by a linear relation until the "pill kink" in 1964. We also see that the errors seem to be correlated: they are all positive or negative during periods of 10 – 20 years. Finally, the linear model is not representative after the pill kink in 1964. In general, it is dangerous to use a fitted model for extrapolation where no predictor variables have been observed (for example: if

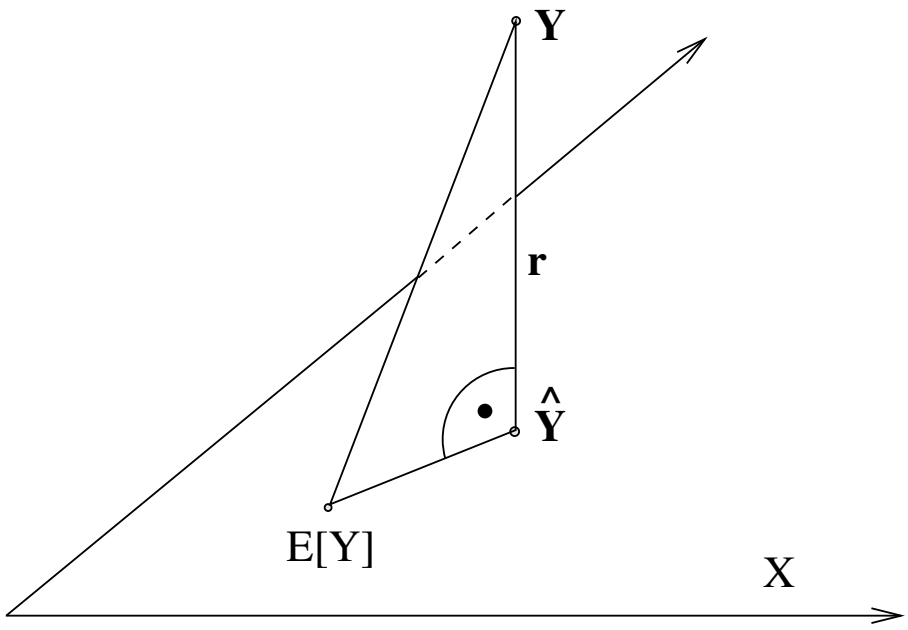


Figure 1.2: The residual vector $\mathbf{r}$ is orthogonal to $\mathcal{X}$.

we would have fitted the linear model in 1964 for prediction of number of births in the future until 2005).

1.3.3 Geometrical Interpretation

The response variable $\mathbf{Y}$ is a vector in $\mathbb{R}^n$. Also, $X\boldsymbol{\beta}$ describes a p-dimensional subspace $\mathcal{X}$ in $\mathbb{R}^n$ (through the origin) when varying $\boldsymbol{\beta} \in \mathbb{R}^p$ (assuming that X has full rank p). The least squares estimator $\widehat{\boldsymbol{\beta}}$ is then such that $X\widehat{\boldsymbol{\beta}}$ is closest to $\mathbf{Y}$ with respect to the Euclidean distance. But this means geometrically that

$$\boxed{X\widehat{\boldsymbol{\beta}} \text{ is the orthogonal projection of } \mathbf{Y} \text{ onto } \mathcal{X}.}$$

We denote the (vector of) fitted values by

$$\hat{\mathbf{Y}} = X\widehat{\boldsymbol{\beta}}.$$

They can be viewed as an estimate of $X\boldsymbol{\beta}$.

The (vector of) residuals is defined by

$$\mathbf{r} = \mathbf{Y} - \hat{\mathbf{Y}}.$$

Geometrically, it is evident that the residuals are orthogonal to $\mathcal{X}$, because $\hat{\mathbf{Y}}$ is the orthogonal projection of $\mathbf{Y}$ onto $\mathcal{X}$. This means that

$$\mathbf{r}^\intercal \mathbf{x}^{(j)} = 0 \text{ for all } j = 1, \ldots, p,$$

where $\mathbf{x}^{(j)}$ is the jth column of X.

We can formally see why the map

$$\mathbf{Y} \mapsto \hat{\mathbf{Y}}$$

is an orthogonal projection. Since $\hat{\mathbf{Y}} = X\widehat{\boldsymbol{\beta}} = X(X^\intercal X)^{-1}X^\intercal \mathbf{Y}$, the map can be represented by the matrix

$$P = X(X^\intercal X)^{-1}X^\intercal. \tag{1.5}$$

It is evident that P is symmetric ($P^\intercal = P$) and P is idem-potent ($P^2 = P$). Furthermore

$$\sum_i P_{ii} = \mathrm{tr}(P) = \mathrm{tr}(X(X^\intercal X)^{-1}X^\intercal) = \mathrm{tr}((X^\intercal X)^{-1}X^\intercal X) = \mathrm{tr}(I_{p\times p}) = p.$$

But these 3 properties characterize that P is an orthogonal projection from $\mathbb{R}^n$ onto a p-dimensional subspace, here $\mathcal{X}$.

The residuals $\mathbf{r}$ can be represented as

$$\mathbf{r} = (I - P)Y,$$

where $I - P$ is now also an orthogonal projection onto the orthogonal complement of $\mathcal{X}$, $\mathcal{X}^\perp = \mathbb{R}^n \setminus \mathcal{X}$, which is $(n-p)$-dimensional. In particular, the residuals are elements of $\mathcal{X}^\perp$.

1.3.4 Don't do many regressions on single variables!

In general, it is not appropriate to replace multiple regression by many single regressions (on single predictor variables). The following (synthetic) example should help to demonstrate this point.

Consider two predictor variables $x^{(1)}, x^{(2)}$ and a response variable Y with the values

$x^{(1)}$	0	1	2	3	0	1	2	3
$x^{(2)}$	−1	0	1	2	1	2	3	4
Y	1	2	3	4	−1	0	1	2

Multiple regression yields the least squares solution which describes the data points exactly

$$Y_i = \hat{Y}_i = 2x_{i1} - x_{i2} \text{ for all } i \qquad (\hat{\sigma}^2 = 0). \tag{1.6}$$

The coefficients 2 and -1, respectively, describe how y is changing when varying either $x^{(1)}$ or $x^{(2)}$ and keeping the other predictor variable constant. In particular, we see that Y decreases when $x^{(2)}$ increases.

On the other hand, if we do a simple regression of Y onto $x^{(2)}$ (while ignoring the values of $x^{(1)}$; and thus, we do not keep them constant), we obtain the least squares estimate

$$\hat{Y}_i = \frac{1}{9}x_{i2} + \frac{4}{3} \text{ for all i} \qquad (\hat{\sigma}^2 = 1.72).$$

This least squares regression line describes how Y changes when varying $x^{(2)}$ while ignoring $x^{(1)}$. In particular, $\hat{Y}$ increases when $x^{(2)}$ increases, in contrast to multiple regression!

The reason for this phenomenon is that $x^{(1)}$ and $x^{(2)}$ are strongly correlated: if $x^{(2)}$ increases, then also $x^{(1)}$ increases. Note that in the multiple regression solution, $x^{(1)}$ has a larger coefficient in absolute value than $x^{(2)}$ and hence, an increase in $x^{(1)}$ has a stronger influence for changing y than $x^{(2)}$. The correlation among the predictors in general makes also the interpretation of the regression coefficients more subtle: in the current setting, the coefficient β_1 quantifies the influence of $x^{(1)}$ on Y *after* having subtracted the effect of $x^{(2)}$ on Y, see also section 1.5.

Summarizing:

> Simple least squares regressions on single predictor variables yield the multiple regression least squares solution, *only* if the predictor variables are orthogonal.
>
> In general, *multiple* regression is the appropriate tool to include effects of more than one predictor variables simultaneously.

The equivalence in case of orthogonal predictors is easy to see algebraically. Orthogonality of predictors means $X^\intercal X = \text{diag}(\sum_{i=1}^n x_{i1}^2, \ldots, \sum_{i=1}^n x_{ip}^2)$ and hence the least squares estimator

$$\hat{\beta}_j = \sum_{i=1}^n x_{ij} Y_i / \sum_{i=1}^n x_{ij}^2 \ (j = 1, \ldots, p),$$

i.e., $\hat{\beta}_j$ depends only on the response variable Y_i and the jth predictor variable x_{ij}.

1.3.5 Computer-Output from R : Part I

We show here parts of the computer output (from R) when fitting a linear model to data about quality of asphalt.

```
   y = LOGRUT  : log("rate of rutting") = log(change of rut depth in inches
                   per million wheel passes)
                   ["rut" := 'Wagenspur', ausgefahrenes Geleise]
  x1 = LOGVISC : log(viscosity of asphalt)
  x2 = ASPH    : percentage of asphalt in surface course
  x3 = BASE    : percentage of asphalt in  base   course
  x4 = RUN     : '0/1' indicator for two sets of runs.
  x5 = FINES   : 10* percentage of fines in surface course
  x6 = VOIDS   : percentage of voids in surface course
```

The following table shows the least squares estimates $\hat{\beta}_j$ $(j = 1, \ldots, 6)$, some empirical quantiles of the residuals r_i $(i = 1, \ldots, n)$, the estimated standard deviation of the errors[2] $\sqrt{\hat{\sigma}^2}$ and the so-called *degrees of freedom* $n - p$.

```
Call:
lm(formula = LOGRUT ~ . , data = asphalt1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.48348 -0.14374 -0.01198  0.15523  0.39652

Coefficients:
             Estimate
(Intercept) -5.781239
LOGVISC     -0.513325
ASPH         1.146898
BASE         0.232809
RUN         -0.618893
```

[2] The term "residual standard error" is a misnomer with a long tradition, since *"standard error"* usually means $\sqrt{\widehat{\text{Var}(\hat{\theta})}}$ for an estimated parameter θ.

```
FINES        0.004343
VOIDS        0.316648

Residual standard error: 0.2604 on 24 degrees of freedom
```

1.4 Properties of Least Squares Estimates

As an introductory remark, we point out that the least squares estimates are random variables: for new data from the same data-generating mechanism, the data would look differently every time and hence also the least squares estimates. Figure 1.3 displays three least squares regression lines which are based on three different realizations from the same data-generating model (i.e., three simulations from a model). We see that the estimates are varying: they are random themselves!

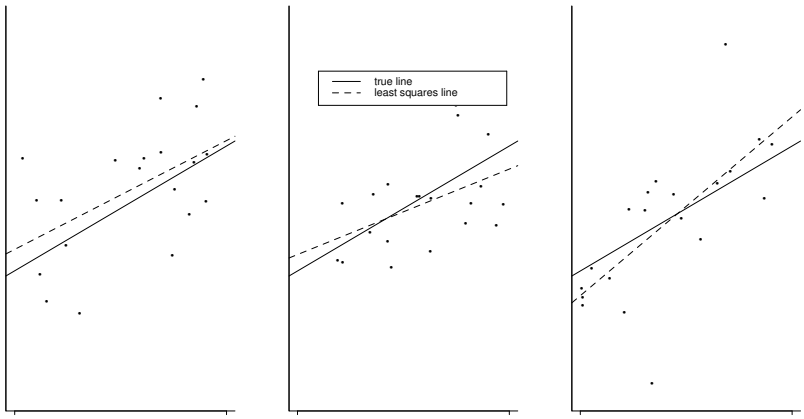


Figure 1.3: Three least squares estimated regression lines for three different data realizations from the same model.

1.4.1 Moments of least squares estimates

We assume here the usual linear model

$$\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},\ \mathbb{E}[\boldsymbol{\varepsilon}] = 0,\ \mathrm{Cov}(\boldsymbol{\varepsilon}) = \mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\intercal] = \sigma^2 I_{n\times n}. \tag{1.7}$$

This means that assumption 1.-4. from section 1.3.2 are satisfied.
It can then be shown that:

(i) $\mathbb{E}[\widehat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$: that is, $\widehat{\boldsymbol{\beta}}$ is **unbiased**

(ii) $\mathbb{E}[\hat{\mathbf{Y}}] = \mathbb{E}[\mathbf{Y}] = X\boldsymbol{\beta}$ which follows from (i). Moreover, $\mathbb{E}[\mathbf{r}] = 0$.

(iii) $\mathrm{Cov}(\widehat{\boldsymbol{\beta}}) = \sigma^2(X^\intercal X)^{-1}$

(iv) $\mathrm{Cov}(\hat{\mathbf{Y}}) = \sigma^2 P$, $\mathrm{Cov}(\mathbf{r}) = \sigma^2(I - P)$

The residuals (which are estimates of the unknown errors ε_i) are also having expectation zero but they are not uncorrelated:

$$\mathrm{Var}(r_i) = \sigma^2(1 - P_{ii}).$$

From this, we obtain

$$\mathbb{E}[\sum_{i=1}^{n} r_i^2] = \sum_{i=1}^{n} \mathbb{E}[r_i^2] = \sum_{i=1}^{n} \mathrm{Var}(r_i) = \sigma^2 \sum_{i=1}^{n} (1 - P_{ii}) = \sigma^2(n - \mathrm{tr}(P)) = \sigma^2(n-p).$$

Therefore, $\mathbb{E}[\hat{\sigma}^2] = \mathbb{E}[\sum_{i=1}^{n} r_i^2/(n-p)] = \sigma^2$ is **unbiased**.

1.4.2 Distribution of least squares estimates assuming Gaussian errors

We assume the linear model as in (1.7) but require in addition that $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d. $\sim \mathcal{N}(0, \sigma^2)$. It can then be shown that:

(i) $\widehat{\boldsymbol{\beta}} \sim \mathcal{N}_p(\boldsymbol{\beta}, \sigma^2 (X^\intercal X)^{-1})$

(ii) $\hat{\mathbf{Y}} \sim \mathcal{N}_n(X\boldsymbol{\beta}, \sigma^2 P), \quad \mathbf{r} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2(I-P))$

(iii) $\hat{\sigma}^2 \sim \frac{\sigma^2}{n-p} \chi^2_{n-p}$.

The normality assumptions of the errors ε_i is often not (approximately) fulfilled in practice. We can then rely on the central limit theorem which implies that for large sample size n, the properties (i)-(iii) above are still approximately true. This is the usual justification in practice to use these properties for constructing confidence intervals and tests for the linear model parameters. However, it is often much better to use **robust methods** in case of non-Gaussian errors which we are not discussing here.

1.5 Tests and Confidence Regions

We assume the linear model as in (1.7) with $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d. $\sim \mathcal{N}(0, \sigma^2)$ (or with ε_i's i.i.d. and "large" sample size n). As we have seen above, the parameter estimates $\widehat{\boldsymbol{\beta}}$ are normally distributed.

If we are interested whether the jth predictor variable is relevant, we can test the null-hypothesis $H_{0,j} : \beta_j = 0$ against the alternative $H_{A,j} : \beta_j \neq 0$. We can then easily derive from the normal distribution of $\hat{\beta}_j$ that

$$\frac{\hat{\beta}_j}{\sqrt{\sigma^2 (X^\intercal X)^{-1}_{jj}}} \sim \mathcal{N}(0,1) \text{ under the null-hypothesis } H_{0,j}.$$

Since σ^2 is unknown, this quantity is not useful, but if we substitute it with the estimate $\hat{\sigma}^2$ we obtain the so-called t-test statistic

$$T_j = \frac{\hat{\beta}_j}{\sqrt{\hat{\sigma}^2 (X^\intercal X)^{-1}_{jj}}} \sim t_{n-p} \text{ under the null-hypothesis } H_{0,j}, \tag{1.8}$$

which has a slightly different distribution than standard Normal $\mathcal{N}(0,1)$. The corresponding test is then called the t-test. In practice, we can thus quantify the relevance of individual predictor variables by looking at the size of the test-statistics T_j $(j = 1, \ldots, p)$ or at the corresponding P-values which may be more informative.

The problem by looking at *individual* tests $H_{0,j}$ is (besides the multiple testing problem in general) that it can happen that all individual tests do not reject the null-hypotheses (say at the 5% significance level) although it is true that some predictor variables have a

significant effect. This "paradox" can occur because of correlation among the predictor variables.

> An individual t-test for $H_{0,j}$ should be interpreted as quantifying the effect of the jth predictor variable after having subtracted the linear effect of all other predictor variables on Y.

To test whether there exists *any* effect from the predictor variables, we can look at the simultaneous null-hypothesis $H_0 : \beta_2 = \ldots = \beta_p = 0$ versus the alternative $H_A : \beta_j \neq 0$ for at least one $j \in \{2, \ldots, p\}$; we assume here that the first predictor variable is the constant $X_{i,1} \equiv 1$ (there are $p-1$ (non-trivial) predictor variables). Such a test can be developed with an analysis of variance (*anova*) decomposition which takes a simple form for this special case:

$$\left\|\mathbf{Y} - \overline{\mathbf{Y}}\right\|^2 = \left\|\hat{\mathbf{Y}} - \overline{\mathbf{Y}}\right\|^2 + \left\|\mathbf{Y} - \hat{\mathbf{Y}}\right\|^2$$

which decomposes the total squared error $\mathbf{Y} - \overline{\mathbf{Y}}$ around the mean $\overline{\mathbf{Y}} = n^{-1}\sum_{i=1}^n Y_i \cdot \mathbf{1}$ as a sum of the squared error due to the regression $\hat{\mathbf{Y}} - \overline{\mathbf{Y}}$ (the amount that the fitted values vary around the global arithmetic mean) and the squared residual error $\mathbf{r} = \mathbf{Y} - \hat{\mathbf{Y}}$. (The equality can be seen most easily from a geometrical point of view: the residuals $\mathbf{r}$ are orthogonal to $\mathcal{X}$ and hence to $\hat{\mathbf{Y}} - \overline{\mathbf{Y}}$). Such a decomposition is usually summarized by an ANOVA table (**AN**alysis **O**f **VA**riance).

	sum of squares	degrees of freedom	mean square	$\mathbf{E}$ [mean square]
regression	$\|\hat{\mathbf{Y}} - \overline{\mathbf{Y}}\|^2$	$p-1$	$\|\hat{\mathbf{Y}} - \overline{\mathbf{Y}}\|^2/(p-1)$	$\sigma^2 + \frac{\|\mathbf{E}[\mathbf{Y}] - \mathbf{E}[\overline{\mathbf{Y}}]\|^2}{p-1}$
error	$\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2$	$n-p$	$\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2/(n-p)$	σ^2
total around global mean	$\|\mathbf{Y} - \overline{\mathbf{Y}}\|^2$	$n-1$	–	–

In case of the global null-hypothesis, there is no effect of any predictor variable and hence $\mathbb{E}[\mathbf{Y}] \equiv const. = \mathbb{E}[\overline{\mathbf{Y}}]$: therefore, the expected mean square equals σ^2 under H_0. The idea is now to divide the mean square by the estimate $\hat{\sigma}^2$ to obtain a scale-free quantity: this leads to the so-called F-statistic

$$F = \frac{\|\hat{\mathbf{Y}} - \overline{\mathbf{Y}}\|^2/(p-1)}{\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2/(n-p)} \sim F_{p-1,n-p} \text{ under the global null-hypothesis } H_0.$$

This test is called the F-test (it is one among several other F-tests in regression).

Besides performing a global F-test to quantify the statistical significance of the predictor variables, we often want to describe the goodness of fit of the linear model for explaining the data. A meaningful quantity is the coefficient of determination, abbreviated by R^2,

$$R^2 = \frac{\|\hat{\mathbf{Y}} - \overline{\mathbf{Y}}\|^2}{\|\mathbf{Y} - \overline{\mathbf{Y}}\|^2}$$

which is the proportion of the total variation of $\mathbf{Y}$ around $\overline{\mathbf{Y}}$ which is explained by the regression (see the ANOVA decomposition and table above).

Similarly to the t-tests as in (1.8), one can derive confidence intervals for the unknown parameters β_j:

$$\hat{\beta}_j \pm \sqrt{\hat{\sigma}^2 (X^\intercal X)^{-1}_{jj}} \cdot t_{n-p;1-\alpha/2}$$

is a two-sided confidence interval which covers the true β_j with probability $1-\alpha$; here, $t_{n-p;1-\alpha/2}$ denotes the $1-\alpha/2$ quantile of a t_{n-p} distribution.

1.5.1 Computer-Output from R : Part II

We consider again the dataset from section 1.3.5. We now give the complete list of summary statistics from a linear model fit to the data.

```
Call:
lm(formula = LOGRUT ~ ., data = asphalt1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.48348 -0.14374 -0.01198  0.15523  0.39652

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.781239   2.459179  -2.351 0.027280 *
LOGVISC     -0.513325   0.073056  -7.027 2.90e-07 ***
ASPH         1.146898   0.265572   4.319 0.000235 ***
BASE         0.232809   0.326528   0.713 0.482731
RUN         -0.618893   0.294384  -2.102 0.046199 *
FINES        0.004343   0.007881   0.551 0.586700
VOIDS        0.316648   0.110329   2.870 0.008433 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2604 on 24 degrees of freedom
Multiple R-Squared: 0.9722,     Adjusted R-squared: 0.9653
F-statistic: 140.1 on 6 and 24 DF,  p-value: < 2.2e-16
```

The table displays the standard errors of the estimates $\sqrt{\widehat{\mathrm{Var}}(\hat{\beta}_j)} = \sqrt{\hat{\sigma}^2 (X^\intercal X)^{-1}_{jj}}$, the t-test statistics for the null-hypotheses $H_{0,j}: \beta_j = 0$ and their corresponding two-sided P-values with some abbreviation about strength of significance. Moreover, the R^2 and adjusted R^2 are given and finally also the F-test statistic for the null-hypothesis $H_0: \beta_2 = \ldots = \beta_p = 0$ (with the degrees of freedom) and its corresponding P-value.

1.6 Analysis of residuals and checking of model assumptions

The residuals $r_i = Y_i - \hat{Y}_i$ can serve as an approximation of the unobservable error term ε_i and for checking whether the linear model is appropriate.

1.6.1 The Tukey-Anscombe Plot

The Tukey-Anscombe is a graphical tool: we plot the residuals r_i (on the y-axis) versus the fitted values $\hat{Y}_i$ (on the x-axis). A reason to plot against the fitted values $\hat{Y}_i$ is that the sample correlation between r_i and $\hat{Y}_i$ is always zero.

In the ideal case, the points in the Tukey-Anscombe plot "fluctuate randomly" around the horizontal line through zero: see also Figure 1.4. An often encountered deviation is non-constant variability of the residuals, i.e., an indication that the variance of ε_i increases with the response variable Y_i: this is shown in Figure 1.5 a)–c). If the Tukey-Anscombe plot shows a trend, there is some evidence that the linear model assumption is not correct

(the expectation of the error is not zero which indicates a systematic error): Figure 1.5d) is a typical example.

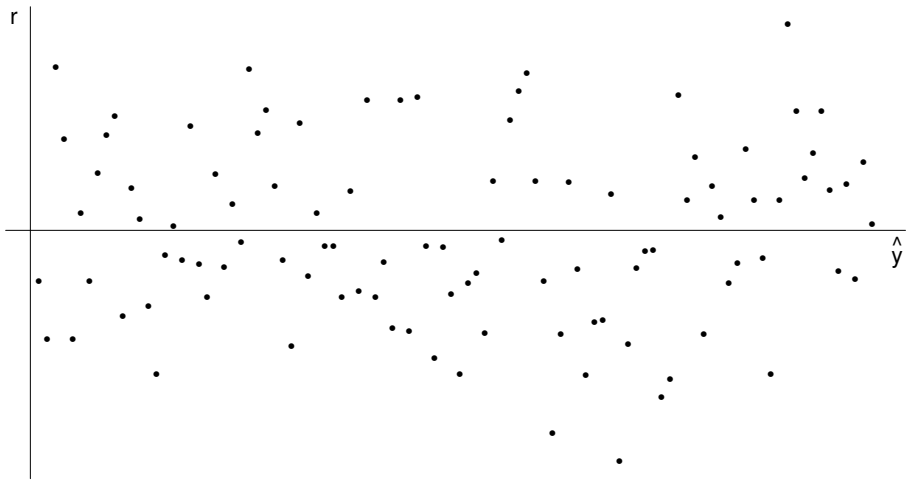


Figure 1.4: Ideal Tukey-Anscombe plot: no violations of model assumptions.

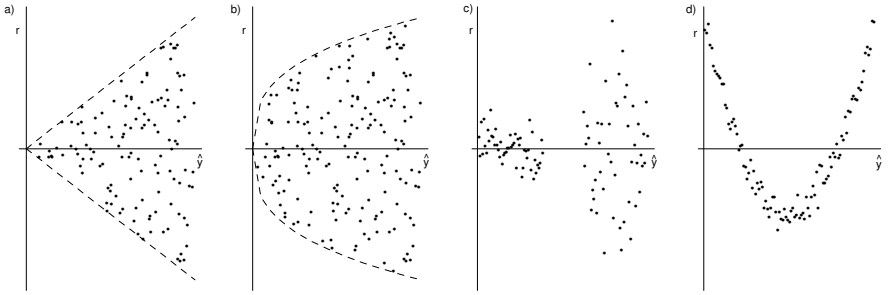


Figure 1.5: a) linear increase of standard deviation, b) nonlinear increase of standard deviation, c) 2 groups with different variances, d) missing quadratic term in the model.

In case where the Tukey-Anscombe plot exhibits a systematic relation of the variability on the fitted values $\hat{Y}_i$, we should either transform the response variable or perform a weighted regression (see Section 1.6.4). If the standard deviation grows linearly with the fitted values (as in Figure 1.5a)), the log-transform $Y \mapsto \log(Y)$ stabilizes the variance; if the standard deviation grows as the square root with the values $\hat{Y}_i$ (as in Figure 1.5b)), the square root transformation $Y \mapsto \sqrt{Y}$ stabilizes the variance.

1.6.2 The Normal Plot

Assumptions for the distribution of random variables can be graphically checked with the QQ (quantile-quantile) plot. In the special case of checking for the normal distribution, the QQ plot is also referred to as a normal plot.

In the linear model application, we plot the empirical quantiles of the residuals (on the y axis) versus the theoretical quantiles of a $\mathcal{N}(0,1)$ distribution (on the x axis). If the residuals would be normally distributed with expectation μ and variance σ^2, the normal plot would exhibit an approximate straight line with intercept μ and slope σ. Figures 1.6 and 1.7 show some normal plots with exactly normally and non-normally distributed observations.

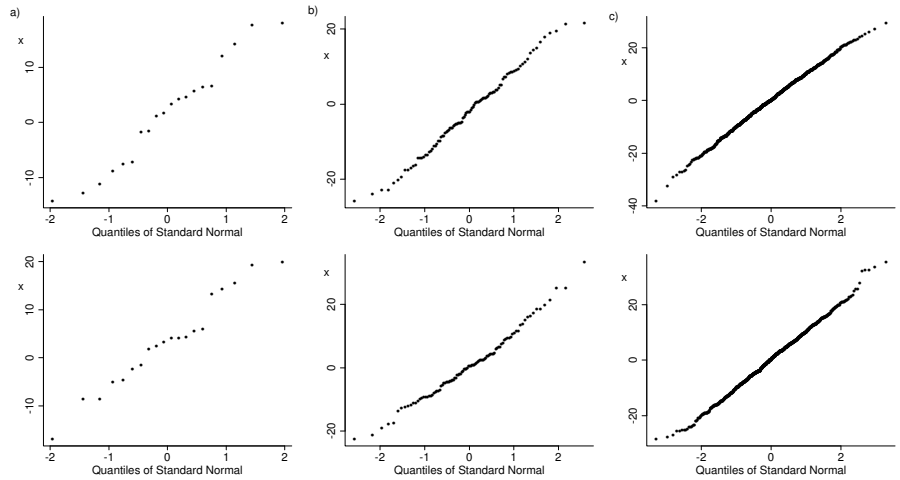


Figure 1.6: QQ-plots for i.i.d. normally distributed random variables. Two plots for each sample size n equal to a) 20, b) 100 and c) 1000.

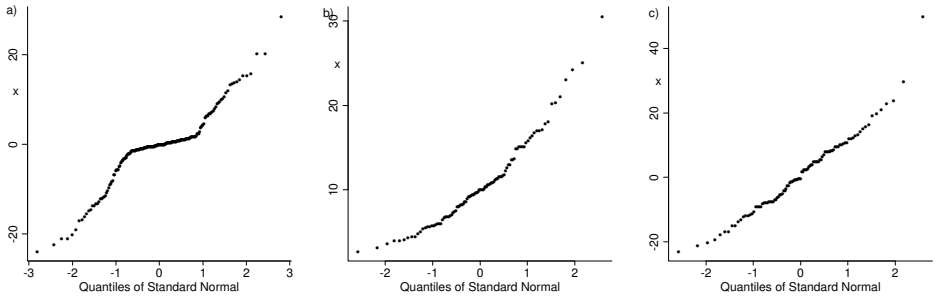


Figure 1.7: QQ-plots for a) long-tailed distribution, b) skewed distribution, c) dataset with outlier.

1.6.3 Plot for detecting serial correlation

For checking independence of the errors we plot the residuals r_i versus the observation number i (or if available, the time t_i of recording the ith observation). If the residuals vary randomly around the zero line, there are no indications for serial correlations among the errors ε_i. On the other hand, if neighboring (with respect to the x-axis) residuals look similar, the independence assumption for the errors seems violated.

1.6.4 Generalized least squares and weighted regression

In a more general situation, the errors are correlated with known covariance matrix,

$$\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \text{with} \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \boldsymbol{\Sigma}).$$

When $\boldsymbol{\Sigma}$ is known (and also in the case where $\boldsymbol{\Sigma} = \sigma^2 G$ with unknown σ^2), this case can be transformed to the i.i.d. one, using a "square root" C such that $\boldsymbol{\Sigma} = CC^\intercal$ (defined, e.g., via Cholesky factorization $\boldsymbol{\Sigma} = LL^\intercal$, L is uniquely determined lower triangular): If $\tilde{\mathbf{Y}} := C^{-1}\mathbf{Y}$ and $\tilde{X} := C^{-1}X$, we have $\tilde{\mathbf{Y}} = \tilde{X}\boldsymbol{\beta} + \tilde{\boldsymbol{\varepsilon}}$, where $\tilde{\boldsymbol{\varepsilon}} \sim \mathcal{N}(\mathbf{0}, I)$. This leads to the *generalized least squares* solution $\widehat{\boldsymbol{\beta}} = (X^\intercal \boldsymbol{\Sigma}^{-1} X)^{-1} X^\intercal \boldsymbol{\Sigma}^{-1} \mathbf{Y}$ with $\mathrm{Cov}(\widehat{\boldsymbol{\beta}}) = (X^\intercal \boldsymbol{\Sigma}^{-1} X)^{-1}$.

A special case where $\mathbf{\Sigma}$ is *diagonal*, $\mathbf{\Sigma} = \sigma^2 \operatorname{diag}(z_1, z_2, \ldots, z_n)$ (with trivial inverse) is the *weighted* least squares problem $\min_{\boldsymbol{\beta}} \sum_{i=1}^n w_i (Y_i - \mathbf{x}_i^\intercal \boldsymbol{\beta})^2$, with weights $w_i \equiv 1/z_i$.

1.7 Model Selection

We assume the linear model

$$Y_i = \sum_{j=1}^p \beta_j x_{ij} + \varepsilon_i \ (i = 1, \ldots, n),$$

where $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d., $\mathbb{E}[\varepsilon_i] = 0$, $\operatorname{Var}(\varepsilon_i) = \sigma^2$.

Problem: Which of the predictor variables should be used in the linear model?
It may be that not all of the p predictor variables are relevant. In addition, every coefficient has to be estimated and thus is afflicted with variability: the individual variabilities for each coefficient sum up and **the variability of the estimated hyper-plane increases the more predictors are entered into the model, whether they are relevant or not**. The aim is often to look for the **optimal** or **best** - not the true - model.

What we just explained in words can be formalized a bit more. Suppose we are looking for optimizing the prediction

$$\sum_{r=1}^q \hat{\beta}_{j_r} x_{ij_r}$$

which includes q predictor variables with indices $j_1, \ldots, j_q \in \{1, \ldots, p\}$. The average mean squared error of this prediction is

$$\begin{aligned} & n^{-1} \sum_{i=1}^n \mathbb{E}[(m(\mathbf{x}_i) - \sum_{r=1}^q \hat{\beta}_{j_r} x_{ij_r})^2] \\ = \ & n^{-1} \sum_{i=1}^n (\mathbb{E}[\sum_{r=1}^q \hat{\beta}_{j_r} x_{ij_r}] - m(\mathbf{x}_i))^2 + \underbrace{n^{-1} \sum_{i=1}^n \operatorname{Var}(\sum_{r=1}^q \hat{\beta}_{j_r} x_{ij_r})}_{= \frac{q}{n}\sigma^2}, \end{aligned} \tag{1.9}$$

where $m(\cdot)$ denotes the regression function in the full model with all the predictor variables. It is plausible that the systematic error (squared bias) $n^{-1} \sum_{i=1}^n (\mathbb{E}[\sum_{r=1}^q \hat{\beta}_{j_r} x_{ij_r}] - m(\mathbf{x}_i))^2$ decreases as the number of predictor variables q increases (i.e., with respect to bias, we have nothing to lose by using as many predictors as we can), but the variance term increases linearly in the number of predictors q (the variance term equals $q/n \cdot \sigma^2$ which is not too difficult to derive). This is the so-called **bias-variance trade-off** which is present in very many other situations and applications in statistics. Finding the best model thus means to optimize the bias-variance trade-off: this is sometimes also referred to as "regularization" (for avoiding a too complex model).

1.7.1 Mallows C_p statistic

The mean squared error in (1.9) is unknown: we do not know the magnitude of the bias term but fortunately, we can estimate the mean squared error.

Denote by $SSE(\mathcal{M})$ the residual sum of squares in a model $\mathcal{M}$: it is overly optimistic and not a good measure to estimate the mean squared error in (1.9). For example,

$SSE(\mathcal{M})$ becomes smaller the bigger the model $\mathcal{M}$ and the biggest model under consideration has the lowest SSE (which generally contradicts the equation in (1.9)).

For any (sub-)model $\mathcal{M}$ which involves some (or all) of the predictor variables, the mean squared error in (1.9) can be estimated by

$$n^{-1}SSE(\mathcal{M}) - \hat{\sigma}^2 + 2\hat{\sigma}^2|\mathcal{M}|/n,$$

where $\hat{\sigma}^2$ is the error variance estimate in the full model and $SSE(\mathcal{M})$ is the residual sum of squares in the submodel $\mathcal{M}$. (A justification can be found in the literature). Thus, in order to estimate the best model, we could search for the sub-model $\mathcal{M}$ minimizing the above quantity. Since $\hat{\sigma}^2$ and n are constants with respect to submodels $\mathcal{M}$, we can also consider the well-known C_p statistic

$$C_p(\mathcal{M}) = \frac{SSE(\mathcal{M})}{\hat{\sigma}^2} - n + 2|\mathcal{M}|$$

and search for the sub-model $\mathcal{M}$ minimizing the C_p statistic.

Other popular criteria to estimate the predictive potential of an estimated model are Akaike's information criterion (AIC) and the Bayesian information criterion (BIC).

Searching for the best model with respect to C_p

If the full model has p predictor variables, there are $2^p - 1$ sub-models (every predictor can be in or out but we exclude the sub-model $\mathcal{M}$ which corresponds to the empty set).

Therefore, an exhaustive search for the sub-model $\mathcal{M}$ minimizing the C_p statistic is only feasible if p is less than say 16 ($2^{16} - 1 = 65'535$ which is already fairly large). If p is "large", we can proceed with stepwise algorithms.

Forward selection.
1. Start with the smallest model $\mathcal{M}_0$ (location model) as the current model.
2. Include the predictor variable to the current model which reduces the residual sum of squares most.
3. Continue step 2. until all predictor variables have been chosen or until a large number of predictor variables has been selected. This produces a sequence of sub-models $\mathcal{M}_0 \subseteq \mathcal{M}_1 \subseteq \mathcal{M}_2 \subseteq \ldots$
4. Choose the model in the sequence $\mathcal{M}_0 \subseteq \mathcal{M}_1 \subseteq \mathcal{M}_2 \subseteq \ldots$ which has smallest C_p statistic.

Backward selection.
1. Start with the full model $\mathcal{M}_0$ as the current model.
2. Exclude the predictor variable from the current model which increases the residual sum of squares the least.
3. Continue step 2. until all predictor variables have been deleted (or a large number of predictor variables). This produces a sequence of sub-models $\mathcal{M}_0 \supseteq \mathcal{M}_1 \supseteq \mathcal{M}_2 \supseteq \ldots$
4. Choose the model in the sequence $\mathcal{M}_0 \supseteq \mathcal{M}_1 \supseteq \mathcal{M}_2 \supseteq \ldots$ which has smallest C_p statistic.

Backward selection is typically a bit better than forward selection but it is computationally more expensive. Also, in case where $p \geq n$, we don't want to fit the full model and forward selection is an appropriate way to proceed.

Chapter 2

Nonparametric Density Estimation

2.1 Introduction

For a moment, we will go back to simple data structures: we have observations which are realizations of univariate random variables,

$$X_1, \ldots, X_n \text{ i.i.d. } \sim F,$$

where F denotes an unknown cumulative distribution function. The goal is to estimate the distribution F. In particular, we are interested in estimating the density $f = F'$, assuming that it exists.

Instead of assuming a parametric model for the distribution (e.g. Normal distribution with unknown expectation and variance), we rather want to be "as general as possible": that is, we only assume that the density exists and is suitably smooth (e.g. differentiable). It is then possible to estimate the unknown density **function** $f(\cdot)$. Mathematically, a function is an **infinite-dimensional** object. Density estimation will become a "basic principle" how to do estimation for infinite-dimensional objects. We will make use of such a principle in many other settings such as nonparametric regression with one predictor variable (Chapter 3) and flexible regression and classification methods with many predictor variables (Chapter 7).

2.2 Estimation of a density

We consider the data which records the duration of eruptions of "Old Faithful", a famous geysir in Yellowstone National Park (Wyoming, USA). You can watch it via web-cam on http://www.nps.gov/yell/oldfaithfulcam.htm

2.2.1 Histogram

The histogram is the oldest and most popular density estimator. We need to specify an *"origin"* x_0 and the *class width* h for the specifications of the intervals

$$I_j = (x_0 + j \cdot h, x_0 + (j+1)h] \quad (j = \ldots, -1, 0, 1, \ldots)$$

for which the histogram counts the number of observations falling into each I_j: we then plot the histogram such that the area of each bar is proportional to the number of observations falling into the corresponding class (interval I_j).

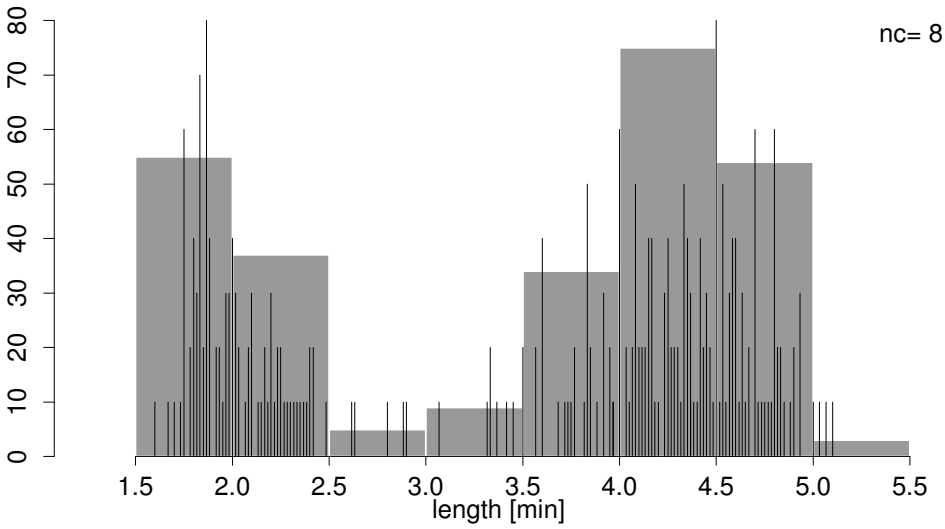
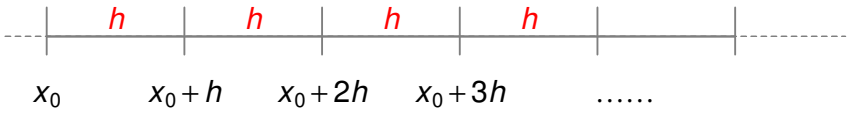


Figure 2.1: Histograms (different class widths) for durations of eruptions of "Old Faithful" geysir in Yellowstone Park ($n = 272$, `data(faithful)`).



The choice of the "origin" x_0 is highly arbitrary, whereas the role of the class width is immediately clear for the user. The form of the histogram depends very much on these two tuning parameters.

2.2.2 Kernel estimator

The naive estimator

Similar to the histogram, we can compute the relative frequency of observations falling into a small region. The density function $f(\cdot)$ at a point x can be represented as

$$f(x) = \lim_{h \to 0} \frac{1}{2h} \mathbb{P}[x - h < X \leq x + h]. \tag{2.1}$$

The naive estimator is then constructed without taking the limit in (2.1) and by replacing probabilities with relative frequencies:

$$\hat{f}(x) = \frac{1}{2hn} \, \#\{i; X_i \in (x - h, x + h]\}. \tag{2.2}$$

This naive estimator is only piecewise constant since every X_i is either in or out of the interval $(x - h, x + h]$. As for histograms, we also need to specify the so-called bandwidth h; but in contrast to the histogram, we do not need to specify an origin x_0.

An alternative representation of the naive estimator (2.2) is as follows. Define the weight function

$$w(x) = \begin{cases} 1/2 & \text{if } |x| \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

Then,

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} w\left(\frac{x - X_i}{h}\right).$$

If we choose instead of the rectangle weight function $w(\cdot)$ a general, typically more smooth kernel function $K(\cdot)$, we have the definition of the kernel density estimator

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right),$$
$$K(x) \geq 0, \ \int_{-\infty}^{\infty} K(x)dx = 1, \ K(x) = K(-x). \tag{2.3}$$

The estimator depends on the *bandwidth* $h > 0$ *which acts as a tuning parameter*. For large bandwidth h, the estimate $\hat{f}(x)$ tends to be very slowly varying as a function of x, while small bandwidths will produce a more wiggly function estimate. The positivity of the kernel function $K(\cdot)$ guarantees a positive density estimate $\hat{f}(\cdot)$ and the normalization $\int K(x)dx = 1$ implies that $\int \hat{f}(x)dx = 1$ which is necessary for $\hat{f}(\cdot)$ to be a density. Typically, the kernel function $K(\cdot)$ is chosen as a probability density which is symmetric around 0.

The smoothness of $\hat{f}(\cdot)$ is inherited from the smoothness of the kernel: if the rth derivative $K^{(r)}(x)$ exists for all x, then $\hat{f}^{(r)}(x)$ exists as well for all x (easy to verify using the chain rule for differentiation).

Popular kernels are the Gaussian kernel

$$K(x) = \varphi(x) = (2\pi)^{-\frac{1}{2}} e^{-x^2/2} \text{ (the density of the } \mathcal{N}(0,1) \text{ distribution)}$$

or a kernel with finite support such as $K(x) = \frac{\pi}{4}\cos(\frac{\pi}{2}x)\mathbf{1}(|x| \leq 1)$. The Epanechnikov kernel, which is optimal with respect to mean squared error, is

$$K(x) = \frac{3}{4}\left(1 - |x|^2\right)\mathbf{1}(|x| \leq 1).$$

But far more important than the kernel is the bandwidth h, see figure 2.2: its role and how to choose it are discussed below.

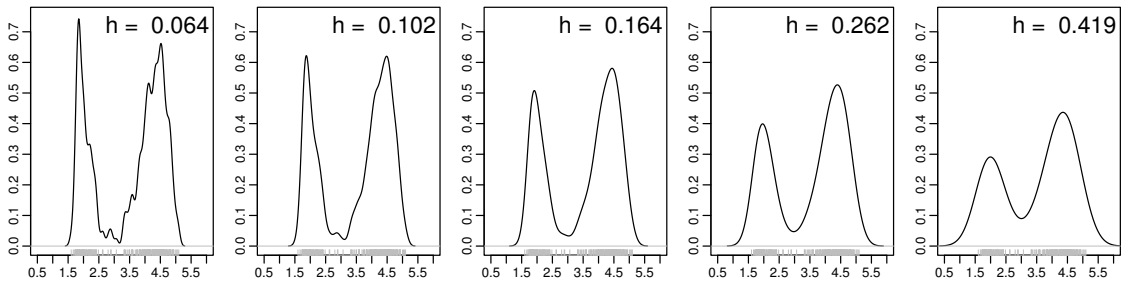


Figure 2.2: *kernel density estimates of the "Old Faithful" eruption lengths; Gaussian kernel and bandwidths* $h = 0.04 \cdot 1.6^{1,2,\ldots,5}$.

2.3 The role of the bandwidth

The bandwidth h is often also called the "smoothing parameter": a moment of thought will reveal that for $h \to 0$, we will have "δ-spikes" at every observation X_i, whereas $\hat{f}(\cdot)$ becomes smoother as h is increasing.

2.3.1 Variable bandwidths: k nearest neighbors

Instead of using a global bandwidth, we can use locally changing bandwidths $h(x)$.

The general idea is to use a large bandwidth for regions where the data is sparse. The k-nearest neighbor idea is to choose

$$h(x) = \text{ Euclidean distance of } x \text{ to the } k\text{th nearest observation,}$$

where k is regulating the magnitude of the bandwidth. Note that generally, $\hat{f}(\cdot)$ will not be a density anymore since the integral is not necessarily equal to one.

2.3.2 The bias-variance trade-off

We can formalize the behavior of $\hat{f}(\cdot)$ when varying the bandwidth h in terms of bias and variance of the estimator. It is important to understand heuristically that

the (absolute value of the) bias of $\hat{f}$ increases and the variance of $\hat{f}$ decreases as h increases.

Therefore, if we want to minimize the mean squared error $\mathrm{MSE}(x)$ at a point x,

$$\mathrm{MSE}(x) = \mathbb{E}\left[\left(\hat{f}(x) - f(x)\right)^2\right] = \left(\mathbb{E}[\hat{f}(x)] - f(x)\right)^2 + \mathrm{Var}(\hat{f}(x)),$$

we are confronted with a **bias-variance trade-off**. As a consequence, this allows - at least conceptually - to optimize the bandwidth parameter (namely to minimize the mean squared error) in a well-defined, coherent way. Instead of optimizing the mean squared error at a point x, one may want to optimize the integrated mean squared error (IMSE)

$$\mathrm{IMSE} = \int \mathrm{MSE}(x)dx$$

which yields an integrated decomposition of squared bias and variance (integration is over the support of X). Since the integrand is non-negative, the order of integration (over the support of X and over the probability space of X) can be reversed, denoted as MISE (mean integrated squared error) and written as

$$\mathrm{MISE} = \mathbb{E}[\int \left(\hat{f}(x) - f(x)\right)^2 dx] \tag{2.4}$$

2.3.3 Asymptotic bias and variance

It is straightforward (using definitions) to give an expression for the exact bias and variance:

$$\begin{aligned}
&\mathbb{E}[\hat{f}(x)] = \int \frac{1}{h} K\left(\frac{x-y}{h}\right) f(y)dy \\
&\mathrm{Var}(\hat{f}(x)) \\
&= \frac{1}{nh^2}\,\mathrm{Var}\left(K\left(\frac{x-X_i}{h}\right)\right) = \frac{1}{nh^2}\mathbb{E}\left[K\left(\frac{x-X_i}{h}\right)^2\right] - \frac{1}{nh^2}\mathbb{E}\left[K\left(\frac{x-X_i}{h}\right)\right]^2 \\
&= n^{-1}\int \frac{1}{h^2} K\left(\frac{x-y}{h}\right)^2 f(y)dy - n^{-1}\left(\int \frac{1}{h} K\left(\frac{x-y}{h}\right) f(y)dy\right)^2.
\end{aligned} \tag{2.5}$$

For the bias we therefore get

$$\begin{aligned}\text{Bias}(x) &= \int \frac{1}{h} K\left(\frac{x-y}{h}\right) f(y)dy - f(x) \\ &\underbrace{=}_{\text{change of variable}} \int K(z) f(x-hz)dz - f(x) = \int K(z)\left(f(x-hz) - f(x)\right) dz. \quad (2.6)\end{aligned}$$

To approximate this expression in general, we invoke an asymptotic argument. We assume that $h \to 0$ as sample size $n \to \infty$, that is:

$$\boxed{h = h_n \to 0 \text{ with } nh_n \to \infty.}$$

This will imply that the bias goes to zero since $h_n \to 0$; the second condition requires that h_n is going to zero more slowly than $1/n$ which turns out to imply that also the variance of the estimator will go to zero as $n \to \infty$. To see this, we use a Taylor expansion of f, assuming that f is sufficiently smooth:

$$f(x-hz) = f(x) - hzf'(x) + \frac{1}{2}h^2z^2f''(x) + \ldots$$

Plugging this into (2.6) yields

$$\begin{aligned}\text{Bias}(x) &= -hf'(x)\underbrace{\int zK(z)dz}_{=0} + \frac{1}{2}h^2 f''(x)\int z^2K(z)dz + \ldots \\ &= \frac{1}{2}h^2 f''(x)\int z^2K(z)dz + \text{higher order terms in } h.\end{aligned}$$

For the variance, we get from (2.5)

$$\begin{aligned}&\text{Var}(\hat{f}(x)) = n^{-1}\int \frac{1}{h^2}K\left(\frac{x-y}{h}\right)^2 f(y)dy - n^{-1}(f(x) + \text{Bias}(x))^2 \\ &= n^{-1}h^{-1}\int f(x-hz)K(z)^2dz - \underbrace{n^{-1}(f(x) + \text{Bias}(x))^2}_{=O(n^{-1})} \\ &= n^{-1}h^{-1}\int f(x-hz)K(z)^2dz + O(n^{-1}) = n^{-1}h^{-1}f(x)\int K(z)^2dz + o(n^{-1}h^{-1}),\end{aligned}$$

assuming that f is smooth and hence $f(x-hz) \to f(x)$ as $h_n \to 0$.
In summary: for $h = h_n \to 0$, $h_n n \to \infty$ as $n \to \infty$

$$\boxed{\begin{array}{lcll}\text{Bias}(x) & = & h^2 f''(x)\int z^2K(z)\,dz/2 + o(h^2) & (n \to \infty) \\ \text{Var}(\hat{f}(x)) & = & (nh)^{-1}f(x)\int K(z)^2dz + o((nh)^{-1}) & (n \to \infty)\end{array}}$$

The optimal bandwidth $h = h_n$ which minimizes the leading term in the asymptotic $\text{MSE}(x)$ can be calculated straightforwardly by solving $\frac{\partial}{\partial h}\text{MSE}(x) = 0$,

$$h_{\text{opt}}(x) = n^{-1/5}\left(\frac{f(x)\int K^2(z)dz}{(f''(x))^2(\int z^2K(z)dz)^2}\right)^{1/5}. \quad (2.7)$$

Since it's not straightforward to estimate and use a *local* bandwidth $h(x)$, one rather considers minimizing the MISE, i.e., $\int MSE(x)\, dx$ which is *asymptotically*

$$\text{asympt. MISE} = \int \text{Bias}(x)^2 \;+\; \text{Var}(\hat{f}(x))\, dx = \frac{1}{4} h^4 R(f'')\, \sigma_K^4 + R(K)/(nh), \tag{2.8}$$

where $R(g) = \int g^2(x)\, dx$, $\sigma_K^2 = \int x^2 K(x)\, dx$, and the "global" asymptotically optimal bandwidth becomes

$$h_{\text{opt}} = n^{-1/5} \left(R(K)/\sigma_K^4 \times 1/R(f'') \right)^{1/5}. \tag{2.9}$$

By replacing h with h_{opt}, e.g., in (2.8), we see that both variance and bias terms are of order $O(n^{-4/5})$, the optimal rate for the MISE and MSE(x). From section 2.4.1, this rate is also optimal for a much larger class of density estimators.

2.3.4 Estimating the bandwidth

As seen from (2.9), the asymptotically best bandwidth depends on $R(f'') = \int f''^2(x)\, dx$ which is unknown (whereas as $R(K)$ and σ_K^2 are known). It is possible to estimate the f'' again by a kernel estimator with an "initial" bandwidth h_{init} (sometimes called a pilot bandwidth) yielding $\hat{f}''_{h_{\text{init}}}$. Plugging this estimate into (2.9) yields an estimated bandwidth $\hat{h}$ for the density estimator $\hat{f}(\cdot)$ (the original problem): of course, $\hat{h}$ depends on the initial bandwidth h_{init}, but choosing h_{init} in an ad-hoc way is less critical for the density estimator than choosing the bandwidth h itself. Furthermore, methods have been devised to determine h_{init} and h simultaneously (e.g., "Sheather-Jones", in R using `density(*, bw="SJ")`).

Estimating local bandwidths

Note that the $h_{\text{opt}}(x)$ bandwidth selection in (2.7) is more problematical mainly because $\hat{f}_{\hat{h}_{\text{opt}}(x)}(x)$ will not integrate to one without further normalization. On the other hand, it can be important to use *locally varying* bandwidths instead of a single global one in a kernel estimator at the expense of being more difficult. The plug-in procedure outlined above can be applied *locally*, i.e., conceptually for each x and hence describes how to estimate local bandwidths from data and how to implement a kernel estimator with locally varying bandwidths. In the related area of nonparametric regression, in section 3.2.1, we will show an example about locally changing bandwidths which are estimated based on an iterative version of the (local) plug-in idea above.

Other density estimators

There are quite a few other approaches to density estimation than the kernel estimators above (whereas in practice, the *fixed* bandwidth kernel estimators are used predominantly because of their simplicity). An important approach in particular aims to estimate the *log density* $\log f(x)$ (setting $\hat{f} = \exp(\widehat{\log f})$) which has no positivity constraints and whose "normal limit" is a simple quadratic. One good implementation is in Kooperberg's R package `logspline`, where spline knots are placed in a stepwise algorithm minimizing approximate BIC (or AIC). This is can be seen as another version of locally varying bandwidths.

2.4 Higher dimensions

Quite many applications involve multivariate data. For simplicity, consider data which are i.i.d. realizations of d-dimensional random variables

$$\mathbf{X}_1, \ldots, \mathbf{X}_n \text{ i.i.d. } \sim f(x_1, \ldots, x_d) dx_1 \cdots dx_d$$

where $f(\cdot)$ denotes the multivariate density.

The multivariate kernel density estimator is, in its simplest form, defined as

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right),$$

where the kernel $K(\cdot)$ is now a function, defined for d-dimensional $\mathbf{x}$, satisfying

$$K(\mathbf{u}) \geq 0, \quad \int_{\mathbb{R}^d} K(\mathbf{u}) d\mathbf{u} = 1, \quad \int_{\mathbb{R}^d} \mathbf{u} K(\mathbf{u}) d\mathbf{u} = \mathbf{0}, \quad \int_{\mathbb{R}^d} \mathbf{u}\mathbf{u}^\intercal K(\mathbf{u}) d\mathbf{u} = I_d.$$

Usually, the kernel $K(\cdot)$ is chosen as a product of a kernel K_{univ} for univariate density estimation

$$K(\mathbf{u}) = \prod_{j=1}^{d} K_{univ}(u_j).$$

If one additionally desires the multivariate kernel $K(\mathbf{u})$ to be *radially symmetric*, it can be shown that K must be the multivariate normal (Gaussian) density, $K(\mathbf{u}) = c_d \exp(-\frac{1}{2}\mathbf{u}^\intercal \mathbf{u})$.

2.4.1 The curse of dimensionality

In practice, multivariate kernel density estimation is often restricted to dimension $d = 2$. The reason is, that a higher dimensional space (with d of medium size or large) will be only very sparsely populated by data points. Or in other words, there will be only very few neighboring data points to any value $\mathbf{x}$ in a higher dimensional space, unless the sample size is extremely large. This phenomenon is also called the *curse of dimensionality*.

An implication of the curse of dimensionality is the following lower bound for the best mean squared error of nonparametric density estimators (assuming that the underlying density is twice differentiable): it has been shown that the best possible MSE rate is

$$O(n^{-4/(4+d)}).$$

The following table evaluates $n^{-4/(4+d)}$ for various n and d:

$n^{-4/(4+d)}$	$d = 1$	$d = 2$	$d = 3$	$d = 5$	$d = 10$
$n = 100$	0.025	0.046	0.072	0.129	0.268
$n = 1000$	0.004	0.010	0.019	0.046	0.139
$n = 100'000$	$1.0\ 10^{-4}$	$4.6\ 10^{-4}$	$13.9\ 10^{-4}$	0.006	0.037

Thus, for $d = 10$, the rate with $n = 100'000$ is still 1.5 times worse than for $d = 1$ and $n = 100$.

Chapter 3

Nonparametric Regression

3.1 Introduction

We consider here nonparametric regression with one predictor variable. Practically relevant generalizations to more than one or two predictor variables are not so easy due to the curse of dimensionality mentioned in section 2.4.1 and often require different approaches, as will be discussed later in Chapter 7.

Figure 3.1 shows (several identical) scatter plots of (x_i, Y_i) $(i = 1, \ldots, n)$. We can model such data as

$$Y_i = m(x_i) + \varepsilon_i, \tag{3.1}$$

where $\varepsilon_1, \ldots, \varepsilon_n$ i.i.d. with $\mathbb{E}[\varepsilon_i] = 0$ and $m : \mathbb{R} \to \mathbb{R}$ is an "arbitrary" function. The function $m(\cdot)$ is called the nonparametric regression function and it satisfies $m(x) = \mathbb{E}[Y|x]$. The restriction we make for $m(\cdot)$ is that it fulfills some kind of smoothness conditions. The regression function in Figure 3.1 does not appear to be linear in x and linear regression is not a good model. The flexibility to allow for an "arbitrary" regression function is very desirable; but of course, such flexibility has its price, namely an inferior estimation accuracy than for linear regression.

3.2 The kernel regression estimator

We can view the regression function in (3.1) as

$$m(x) = \mathbb{E}[Y|X = x],$$

(assuming that X is random and $X_i = x_i$ are realized values of the random variables). We can express this conditional expectation as

$$\int_{\mathbb{R}} y f_{Y|X}(y|x)dy = \frac{\int_{\mathbb{R}} y f_{X,Y}(x, y)dy}{f_X(x)},$$

where $f_{Y|X}$, $f_{X,Y}$, f_X denote the conditional, joint and marginal densities. We can now plug in the univariate and bivariate kernel density (all with the same univariate kernel K) estimates

$$\hat{f}_X(x) = \frac{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)}{nh}, \quad \hat{f}_{X,Y}(x, y) = \frac{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) K\left(\frac{y-Y_i}{h}\right)}{nh^2}$$

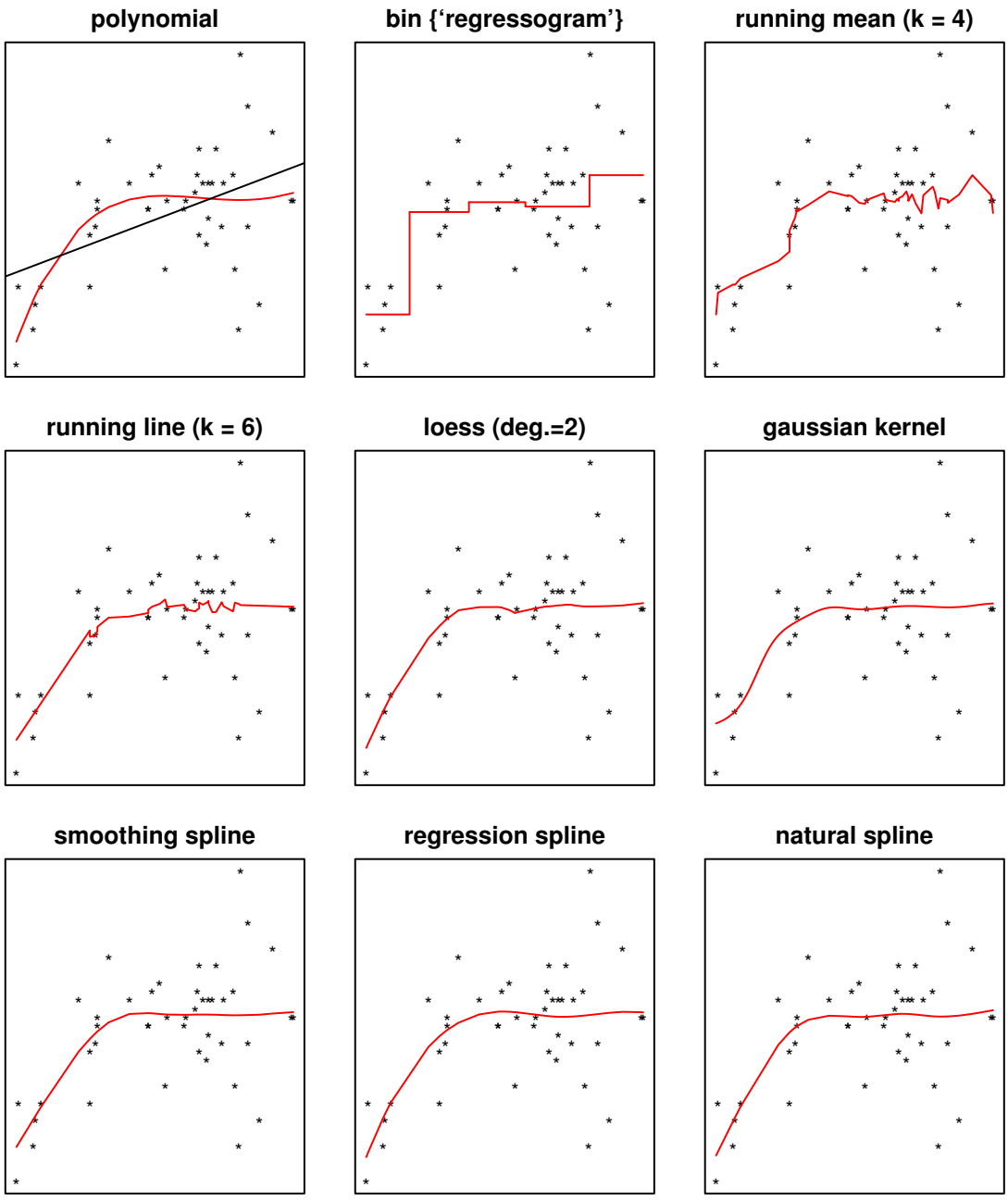


Figure 3.1: Various regression estimators in model $Y_i = m(x_i) + \varepsilon_i$ $(i = 1, \ldots, 43)$ with response Y a log-concentration of a serum (in connection of Diabetes) and predictor variable x the age in months of children. See Hastie and Tibshirani (1990, p.10). Except for the linear regression fit (top left panel), all other estimators have about 5 degrees of freedom.

into the formula above which yields the so-called Nadaraya-Watson kernel estimator

$$\hat{m}(x) = \frac{\sum_{i=1}^{n} K((x - x_i)/h) Y_i}{\sum_{i=1}^{n} K((x - x_i)/h)} = \frac{\sum_{i=1}^{n} \omega_i Y_i}{\sum_i \omega_i}, \tag{3.2}$$

i.e., a weighted mean of the Y_i where $\omega_i = \omega_i(x)$ is a kernel centered at x_i. An interesting interpretation of the kernel regression estimator in (3.2) is

$$\hat{m}(x) = \arg\min_{m_x \in \mathbb{R}} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)(Y_i - m_x)^2. \tag{3.3}$$

This can be easily verified by solving $\frac{d}{dm_x}\sum_{i=1}^{n} K((x-x_i)/h)(Y_i - m_x)^2 = 0$. Thus, for every fixed x, we are searching for the best local constant m_x such that the localized sum of squares is minimized; localization is here described by the kernel and gives a large weight to those observations (x_i, Y_i) where x_i is close to the point x of interest.

3.2.1 The role of the bandwidth

Analogously as in section 2.3, the bandwidth h controls the bias-variance trade-off: a large bandwidth h implies high bias but small variance, resulting in a slowly varying curve, and vice-versa. We are not showing the computations for MSE(x), just note that they not only depend on (derivatives of) $m(x)$, but also on $f_X(x)$.

Local bandwidth selection

Similarly as in (2.7), also using $\int uK(u)du = 0$, there is a formula of the asymptotically best local bandwidth $h_{\text{opt}}(x)$ which depends on $m''(\cdot)$ and the error variance σ_ε^2:

$$h_{\text{opt}}(x) = n^{-1/5}\left(\frac{\sigma_\varepsilon^2 \int K^2(z)dz}{\{m''(x)\int z^2 K(z)dz\}^2}\right)^{1/5}. \tag{3.4}$$

The locally optimal bandwidth $h_{\text{opt}}(x)$ can then be estimated in an iterative way using

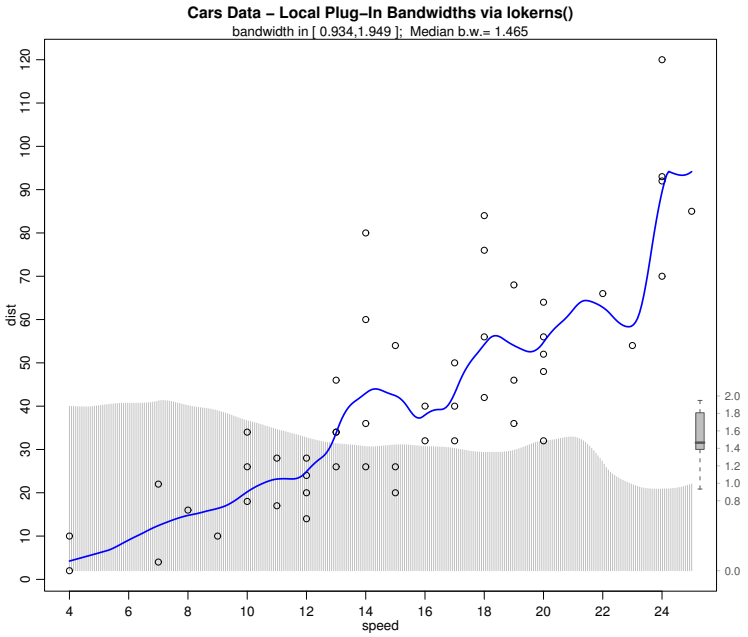


Figure 3.2: Nonparametric function estimate and locally varying bandwidths for distance of stopping as a function of speed of cars.

the plug-in principle. Roughly speaking, start with an initial bandwidth h_0 to estimate

$m''(\cdot)$ (by using an inflated version $n^{1/10}h_0$) and σ_ε^2; these estimates can then be used to get a first estimate of $h_{\text{opt}}(x)$. Now use this first bandwidth estimate as the current bandwidth h_1 to estimate again $m''(\cdot)$ (by using the inflated version $n^{1/10}h_1$) and σ_ε^2, and then obtain new estimates for $h_{\text{opt}}(x)$; and so on, see Brockmann et al. (1993).

Such a procedure has been implemented in R with the function `lokerns` in the package `lokern`. The dataset `cars` contains the distance for stopping as a function of speed of a car. A nonparametric function estimate with locally varying bandwidth can then be obtained as follows:

```
library(lokern); lofit <- lokerns(cars$ speed, cars$ dist)
```

3.2.2 Inference for the underlying regression curve

We consider here the properties, in particular the variability, of the kernel estimator $\hat{m}(x_i)$ at an observed design point x_i.

The hat matrix

It is useful to represent the kernel estimator evaluated at the design points $\hat{m}(x_1), \ldots, \hat{m}(x_n)$ as a *linear* operator (on $\mathbb{R}^n$, i.e., a matrix):

$$\begin{aligned} \mathcal{S}: \qquad & \mathbb{R}^n \to \mathbb{R}^n, \\ & (Y_1, \ldots, Y_n)^\intercal \mapsto (\hat{m}(x_1), \ldots, \hat{m}(x_n))^\intercal =: \hat{\mathbf{m}}(x_\cdot) = \hat{\mathbf{Y}}, \end{aligned}$$

i.e., $\hat{\mathbf{Y}} = \mathcal{S}\mathbf{Y}$ where $\mathcal{S}$ is the matrix representing the linear operator above. The kernel estimator in (3.2) is of the form

$$\hat{m}(x) = \sum_{i=1}^n w_i(x) Y_i, \;\; w_i(x) = \frac{K((x - x_i)/h)}{\sum_{j=1}^n K((x - x_j)/h)}.$$

Therefore, the matrix $\mathcal{S}$ which represents the operator above is

$$[\mathcal{S}]_{r,s} = w_s(x_r), \;\; r, s \in \{1, \ldots, n\},$$

since $\mathcal{S}\{(Y_1, \ldots, Y_n)^\intercal\} = (\hat{m}(x_1), \ldots, \hat{m}(x_n))^\intercal$. The "smoother" matrix $\mathcal{S}$ is also called the "*hat matrix*", since it yields the vector of fitted values (at the observed design points x_i). Note that many other nonparametric regression methods (including those in the next two sections) can be seen to be linear in $\mathbf{Y}$ and hence be written as $\hat{\mathbf{Y}} = \mathcal{S}\mathbf{Y}$ where $\mathcal{S}$ depends on the x-design $(x_1, x_2, \ldots, x_n)$ and typically a smoothing parameter, say h. Algorithmically, for $\hat{\mathbf{Y}} = s(\mathbf{x}, \mathbf{Y}, h)$, the hat matrix can easily be computed columnwise as $\mathcal{S}_{\cdot,j} = s(\mathbf{x}, \mathbf{e}_j, h)$ where $\mathbf{e}_j$ is the unit vector with $(\mathbf{e}_j)_i = \delta_{i,j} := \mathbf{1}(i = j)$.

Because of the elementary formula $\text{Cov}(A\mathbf{X}) = A\,\text{Cov}(\mathbf{X})A^\intercal$ (for a non-random matrix A and random vector $\mathbf{X}$), we get the covariance matrix

$$\text{Cov}(\hat{\mathbf{m}}(x_\cdot)) = \sigma_\varepsilon^2 \mathcal{S}\mathcal{S}^\intercal, \tag{3.5}$$

i.e., $\text{Cov}(\hat{m}(x_i), \hat{m}(x_j)) = \sigma_\varepsilon^2(\mathcal{S}\mathcal{S}^\intercal)_{ij}$, and $\text{Var}(\hat{m}(x_i)) = \sigma_\varepsilon^2(\mathcal{S}\mathcal{S}^\intercal)_{ii}$.

Degrees of freedom

One way to assign degrees of freedom for regression estimators with a linear hat-operator $\mathcal{S}$ is given by

$$df = \text{trace}(\mathcal{S}). \tag{3.6}$$

This definition coincides with the notion we have seen in the linear model: there, (1.5), the fitted values $\hat{Y}_1, \ldots, \hat{Y}_n$ can be represented by the projection $P = X(X^\intercal X)^{-1}X^\intercal$, which is the hat matrix, and $\mathrm{trace}(P) = \mathrm{trace}((X^\intercal X)^{-1}X^\intercal X) = \mathrm{trace}(I_p) = p$ equals the number of parameters in the model. Thus, the definition of degrees of freedom above can be viewed as a general concept for the number of parameters in a model fit with linear hat matrix.

Estimation of the error variance

Formula (3.5) requires knowledge of σ_ε^2. A plausible estimate is via the residual sum of squares,

$$\hat{\sigma}_\varepsilon^2 = \frac{\sum_{i=1}^n (Y_i - \hat{m}(x_i))^2}{n - df}.$$

We then get an estimate for the standard error of the kernel regression estimator at the design points via (3.5):

$$\widehat{s.e}(\hat{m}(x_i)) = \sqrt{\widehat{\mathrm{Var}}(\hat{m}(x_i))} = \sqrt{\frac{\sum_{j=1}^n (Y_j - \hat{m}(x_j))^2}{n - df}(\mathcal{S}\mathcal{S}^\intercal)_{ii}}.$$

The estimated standard errors above are useful since under regularity conditions, $\hat{m}(x_i)$ is asymptotically normal distributed:

$$\hat{m}(x_i) \approx \mathcal{N}(\mathbb{E}[\hat{m}(x_i)], \mathrm{Var}(\hat{m}(x_i))),$$

so that

$$I = \hat{m}(x_i) \pm 1.96 \cdot \widehat{s.e.}(\hat{m}(x_i))$$

yields approximate pointwise confidence intervals for $\mathbb{E}[\hat{m}(x_i)]$. Some functions in R (e.g. the function `gam` from package `mgcv`, see Chapter 7) supply such pointwise confidence intervals. Unfortunately, it is only a confidence interval for the expected value $\mathbb{E}[\hat{m}(x_i)]$ and not for the true underlying function $m(x_i)$. Correction of this interval is possible by subtracting a bias estimate: i.e., instead of the interval I above, we can use $I - \widehat{\mathrm{bias}}$, where $\widehat{\mathrm{bias}}$ is an estimate of the bias (which is not so easy to construct; see also section 2.3).

3.3 Local polynomial nonparametric regression estimator

As a starting point, consider the kernel estimator which can be represented as a locally constant function as in (3.3). This can now be extended to functions which are locally polynomial. We aim to find local regression parameters $\boldsymbol{\beta}(x)$, defined as

$$\widehat{\boldsymbol{\beta}}(x) = \operatorname*{arg\,min}_{\boldsymbol{\beta} \in \mathbb{R}^p} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \left(Y_i - \beta_1 - \beta_2(x_i - x) - \ldots - \beta_p(x_i - x)^{p-1}\right)^2.$$

An even number p turns out to be better: in practice, we often choose $p = 2$ or $p = 4$. The estimated local regression parameter $\widehat{\boldsymbol{\beta}}(x)$ describes a local polynomial regression fit, localized and centered at x. The function estimator is then given by evaluating this local regression fit $\sum_{j=1}^p \hat{\beta}_j(x)(u-x)^{j-1}$ at $u = x$: due to the centering, only the local intercept remains and the local polynomial function estimator becomes

$$\hat{m}(x) = \hat{\beta}_1(x).$$

Note that due to (local) correlation among the $(x_i - x)^j$'s, $\hat{\beta}_1(x)$ is not the same as a local constant fit from (3.3).

The local polynomial estimator is often better at the edges than the locally constant Nadaraya-Watson kernel estimator. Another interesting property is that the method also immediately yields estimates for the derivatives of the function: when differentiating the local regression fit $\sum_{j=1}^{p} \hat{\beta}_j(x)(u-x)^{j-1}$ with respect to u and evaluating it at x, we obtain

$$\hat{m}^{(r)}(x) = r!\hat{\beta}_{r+1}(x) \ (r = 0, 1, \ldots p-1).$$

3.4 Smoothing splines and penalized regression

Function estimation could also be done by using higher order global polynomials, which is often not advisable, or by using splines which can be specified by choosing a set of knots. The latter is a more locally oriented approach and is called "regression splines". Here, we discuss a method based on splines *without* having to specify where to select the knots of the spline.

3.4.1 Penalized sum of squares

Consider the following problem: among all functions m with continuous second derivatives, find the one which minimizes the penalized residual sum of squares

$$\sum_{i=1}^{n}(Y_i - m(x_i))^2 + \lambda \int m''(z)^2 dz, \tag{3.7}$$

where $\lambda \geq 0$ is a smoothing parameter. The first term measures closeness to the data and the second term penalizes curvature ("roughness") of the function. The two extreme cases are:

- $\lambda = 0$: m is any function interpolating the data (but for $\lambda \to 0$, in the limit, $m_\lambda \to$ the well defined interpolating natural cubic spline).
- $\lambda = \infty$: the least squares fit for linear regression which fulfills $m''(x) \equiv 0$.

Thus, a large λ corresponds to a smooth function.

3.4.2 The smoothing spline solution

Remarkably, the minimizer of (3.7) is *finite*-dimensional, although the criterion to be minimized is over a Sobolev space of functions (function space for which the integral $\int m''^2$ is defined), an infinite-dimensional space. Let us assume for now that the data has x values sorted and unique, $x_1 < x_2 < \ldots < x_n$.

The solution $\hat{m}_\lambda(\cdot)$ (i.e., the unique minimizer of (3.7)) is a natural **cubic spline** with knots at the predictors x_i: that is, $\hat{m}$ is a piecewise cubic polynomial in each interval $[x_i, x_{i+1})$ such that $\hat{m}_\lambda^{(k)}$ $(k = 0, 1, 2)$ is continuous everywhere and ("natural") $\hat{m}''(x_1) = \hat{m}''(x_n) = 0$. For the $n-1$ cubic polynomials, we'd need $(n-1) \cdot 4$ coefficients. Since there are $(n-2) \cdot 3$ continuity conditions (at every "inner knot", $i = 2, \ldots, n-1$) plus the 2 "natural" conditions, this leaves $4(n-1) - [3(n-2)+2] = n$ free parameters (the β_j's

below). Knowing that the solution is a cubic spline, it can be obtained by linear algebra. The trick is to represent

$$\hat{m}_\lambda(x) = \sum_{j=1}^{n} \beta_j B_j(x), \tag{3.8}$$

where the $B_j(\cdot)$'s are basis functions for natural splines. The unknown coefficients can then be estimated from least squares in linear regression under side constraints. The criterion in (3.7) for $\hat{m}_\lambda$ as in (3.8) then becomes

$$\|Y - B\boldsymbol{\beta}\|^2 + \lambda \boldsymbol{\beta}^\intercal \Omega \boldsymbol{\beta},$$

where the design matrix B has jth column $(B_j(x_1), \ldots, B_j(x_n))^\intercal$ and $\Omega_{jk} = \int B_j''(z) B_k''(z) dz$. The solution can then be derived in a straightforward way,

$$\widehat{\boldsymbol{\beta}}_{n\times 1} = (B^\intercal B + \lambda\Omega)^{-1} B^\intercal Y. \tag{3.9}$$

This can be computed efficiently using fast linear algebra, particularly when B is a banded matrix.

The fitted values are then $\hat{Y}_i = \hat{m}_\lambda(x_i)$ $(i = 1, \ldots, n)$ and

$$(\hat{Y}_1, \ldots, \hat{Y}_n)^\intercal = \mathcal{S}_\lambda Y, \quad \mathcal{S}_\lambda = B(B^\intercal B + \lambda\Omega)^{-1} B^\intercal. \tag{3.10}$$

The hat matrix $\mathcal{S}_\lambda = \mathcal{S}_\lambda{}^\intercal$ is here symmetric which implies elegant mathematical properties (real-valued eigen-decomposition).

3.4.3 Shrinking towards zero

At first sight, the smoothing spline solution in (3.8) looks heavily over-parameterized since we have to fit n unknown coefficients $\beta_1, \ldots, \beta_n$. However, the solution in (3.9) is not the least squares estimator but rather a Ridge-type version: the matrix $\lambda\Omega$ serves as a Ridge or shrinkage matrix so that the estimates $\hat{\beta}$ are shrunken towards zero: i.e., for large λ, the expression $(B^\intercal B + \lambda\Omega)^{-1}$ becomes small. Thus, since all the coefficients are shrunken towards zero, we gain on the variance part of each $\hat{\beta}_j$ by the square of the shrinkage factor, and the overall smoothing spline fit will be appropriate if λ is chosen suitably.

Note that λ can be chosen on the scale of equivalent degrees of freedom (df): df $=$ $trace(\mathcal{S}_\lambda)$. This provides an intuitive way to specify a smoothing spline: e.g. a smoothing spline with df=5 is as complex as a global polynomial of degree 4 (which has 5 parameters including the intercept), see also Figure 3.1.

3.4.4 Relation to equivalent kernels

It is interesting to note that there is a relationship between the smoothing spline estimate and a particular kernel estimator. The smoothing spline estimate $\hat{m}(x)$ is approximately

$$\begin{aligned}
\hat{m}_\lambda(x) &\approx \sum_{i=1}^{n} w_i(x) Y_i, \\
w_i(x) &= \frac{1}{nh(x) f_X(x)} K\left(\frac{x - x_i}{h(x)}\right), \\
h(x) &= \lambda^{1/4} n^{-1/4} f_X(x)^{-1/4}, \\
K(u) &= \frac{1}{2} \exp\left(-\frac{|u|}{\sqrt{2}}\right) \sin\left(\frac{|u|}{\sqrt{2}} + \frac{\pi}{4}\right).
\end{aligned}$$

See for example Green and Silverman (1994, Ch. 3.7).

The important fact is here that the bandwidth of the equivalent kernel estimator has a *local bandwidth*, depending on the density of the predictor variable x. In regions where the density of the predictor is low (observations are sparse), the bandwidth automatically adapts and becomes large: intuitively, this is the right behavior because we should use strong smoothing in regions where only few observations are available.

An example of a smoothing spline fit for real data is displayed in Figure 3.1. Finally, we illustrate on an artificial dataset the advantage of smoothing splines to adapt to the density of the predictor variables. Figure 3.3 shows the performance of smoothing splines in comparison with the Nadaraya-Watson Gaussian kernel estimator. The data has the following structure:

- the density of the predictor is high for positive values and low for negative values
- the true function is strongly oscillating where the predictor density is high and slowly oscillating where the predictors are sparse

The smoothing spline fit (using the GCV criterion for selecting the degrees of freedom, see section 4.5) yields a very good fit: it captures the strong oscillations because there are many data points with positive values of the predictors. On the other hand, the kernel estimator has been tuned such that it captures the strong oscillations, using a small bandwidth h (this was done by knowing the true underlying function – which is not feasible in practice): but the small bandwidth h then causes a much too rough and poor estimate for negative predictor values, although the underlying true function is smooth.

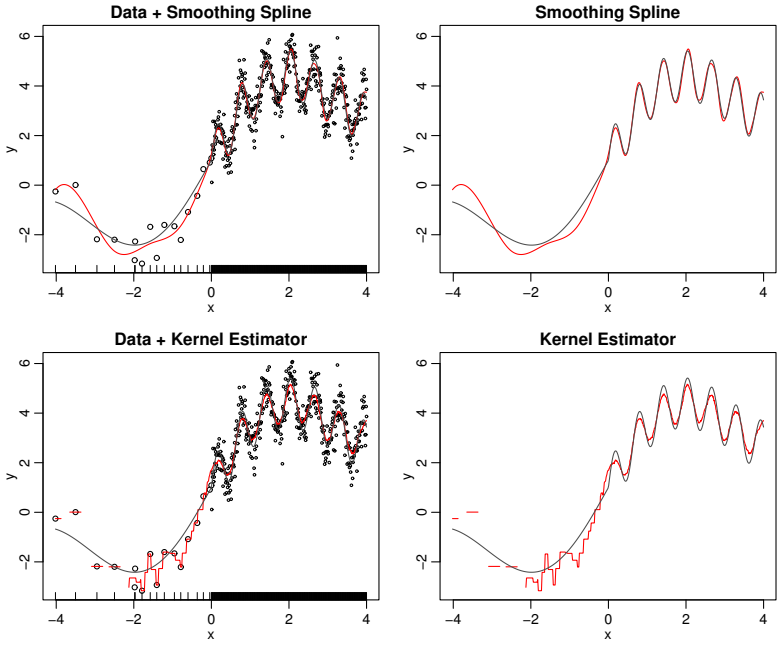


Figure 3.3: Curve estimation for synthetic dataset ($n = 417$). Left panels: scatterplot, overlaid with curve estimates (red) and true curve (gray); Right panels: curve estimates (red) and true curve (gray).
Top: Smoothing spline with GCV-selected df (= 9.23); Bottom: Nadaraya-Watson kernel estimator with bandwidth chosen for good estimation of the strong oscillations of the true function (giving $df = 25.6$).

Chapter 4

Cross-Validation

4.1 Introduction

The generalization performance (which is a terminology from the machine learning community) of a learning method describes its prediction capacity on new test data. Or in other words, the generalization performance measures the predictive power of a learning method on new, out-sample data. Having an estimate of this predictive power is very important for comparing among different methods or for tuning an estimator or algorithm to optimize predictive power.

Cross-Validation is one very general method for estimating such generalization or out-sample performance.

4.2 Training and Test Set

Consider the situation where we have data

$$(X_1, Y_1), \ldots, (X_n, Y_n) \text{ i.i.d. } \sim P$$

where P denotes the unknown distribution. (In case where the x_i is non-random, we can also drop the i.i.d. assumption for the pairs (x_i, Y_i).)

We typically have a target in mind, for example the regression function $m(\cdot)$. The estimated regression function $\hat{m}(\cdot)$ is then constructed by using some estimator or algorithm, based on the data $(X_1, Y_1), \ldots, (X_n, Y_n)$. This data is also called the **training data**, since it is used to train the estimator or learning algorithm.

We then would like to evaluate the accuracy of the estimated target which is based on the training data. A principal problem thereby is that if we use the training data again to measure the predictive power of our estimated target (e.g. of $\hat{m}(\cdot)$), the results will be overly optimistic. For example, when using the residual sum or squares in regression,

$$\sum_{i=1}^{n} (Y_i - \hat{m}(X_i))^2$$

becomes smaller the more "complex" (more degrees of freedom) the model for $\hat{m}(\cdot)$ involves. Obviously, a very complex model will not be good. We have seen already in Section 3.4 that penalizing the residual sum of squares can be useful to cope with the overfitting problem.

Alternatively, we could look how well the estimated target, e.g. $\hat{m}(\cdot)$, does on **new test data**

$$(X_{new,1}, Y_{new,1}), \ldots, (X_{new,\ell}, Y_{new,\ell}) \text{ i.i.d. } \sim P,$$

which is assumed to be independent from the training data but having the same distribution P. In case of regression, we would like to evaluate

$$\ell^{-1} \sum_{i=1}^{\ell} (Y_{new,i} - \hat{m}(X_{new,i}))^2.$$

More generally, when having an estimated target $\hat{m}$ and a loss function ρ, we would like to evaluate

$$\ell^{-1} \sum_{i=1}^{\ell} \rho\left(Y_{new,i}, \hat{m}(X_{new,i})\right),$$

where $\hat{m}(\cdot)$ is constructed from training data only. If ℓ is large, this evaluation on the test set approximates the theoretical test set error

$$\mathbb{E}_{(X_{new},Y_{new})}[\rho(Y_{new}, \underbrace{\hat{m}}_{\text{based on training data only}}(X_{new}))]$$

which is still a function of the training data. The expected value of this (with respect to the training data) is the generalization error,

$$\mathbb{E}_{training}\mathbb{E}_{(X_{new},Y_{new})}[\rho(Y_{new}, \underbrace{\hat{m}}_{\text{...training data}}(X_{new}))] = \mathbb{E}[\rho(Y_{new}, \hat{m}(X_{new}))], \tag{4.1}$$

where the latter $\mathbb{E}$ is over the training data $(X_1, Y_1), \ldots, (X_n, Y_n)$ as well as the test data (X_{new}, Y_{new}). This generalization error avoids the fact that it would get smaller the more complex the model for $\hat{m}(\cdot)$. In particular, it will increase as $\hat{m}$ is overfitting. Hence the generalization error in (4.1) is a very useful quantity to regularize and tune an estimator or algorithm for $\hat{m}$.

4.3 Constructing training-, test-data and cross-validation

Unfortunately, test data is not available at the time we want to run and tune our estimator or algorithm. But we can always "artificially" construct (smaller) training- and test-data.

4.3.1 Leave-one-out cross-validation

For leave-one-out cross-validation (CV), we use the ith sample point as test data (test sample size = 1) and the remaining $n-1$ sample points as training data.

Denote in general the estimator or algorithm by $\hat{\theta}_n$ which is based on the n sample points. In CV, when deleting the ith sample, the estimator is based on the sample without the ith observation, and we denote this by

$$\hat{\theta}_{n-1}^{(-i)}, \ i = 1, \ldots, n.$$

We can then evaluate this estimate on the ith observation (the test sample), for every $i = 1, \ldots, n$. To make this more explicit, we suppose that the estimator $\hat{\theta}$ is a curve

estimator $\hat{m}$, and performance is evaluated in terms of a loss function ρ, e.g. $\rho(u) = u^2$ as in Chapter 3. The cross-validated performance is then

$$n^{-1} \sum_{i=1}^{n} \rho\left(Y_i, \hat{m}_{n-1}^{(-i)}(X_i)\right) \tag{4.2}$$

which is an estimate of the test set error, or generalization error, in (4.1).

Note that the CV-method is very general: it can be used for any loss function ρ and in many problems which can be different than linear or nonparametric regression.

CV in general requires that the estimator $\hat{\theta}$ is fitted n-times, namely for all training sets where the ith observation has been deleted ($i = 1, \ldots, n$).

4.3.2 K-fold Cross-Validation

A computationally cheaper version of leave-one-out CV is the so-called K-fold CV. The data set is randomly partitioned into K equally sized (as equal as possible) subsets $\mathcal{B}_k$ of $\{1, \ldots, n\}$ such that $\cup_{k=1}^{K} \mathcal{B}_k = \{1, \ldots, n\}$ and $\mathcal{B}_j \cap \mathcal{B}_k = \emptyset$ ($j \neq k$). We can now set aside a kth test data set including all sample points whose indices are elements of $\mathcal{B}_k$.

K-fold cross-validation then uses the sample points with indices not in $\mathcal{B}_k$ as training set to construct an estimator

$$\hat{\theta}_{n-|\mathcal{B}_k|}^{(-\mathcal{B}_k)}.$$

The analogue of the evaluation formula in (4.2) is then, for regression with $\hat{m}(\cdot)$,

$$K^{-1} \sum_{k=1}^{K} |\mathcal{B}_k|^{-1} \sum_{i \in \mathcal{B}_k} \rho\left(Y_i, \hat{m}_{n-|\mathcal{B}_k|}^{-(\mathcal{B}_k)}(X_i)\right).$$

K-fold CV thus only needs to run the estimation algorithm K times. Leave-one-out CV is the same as n-fold CV. In practice, often $K = 5$ or $K = 10$ are used.

4.3.3 Random divisions into test- and training-data

K-fold CV has the disadvantage that it depends on the **one** realized random partition into subsets $\mathcal{B}_1, \ldots, \mathcal{B}_K$. In particular, if the data (pairs) are assumed to be i.i.d., the indexing of the data (pairs) should not have an influence on validating a performance: note that leave-one-out CV is indeed independent of indexing the data.

In principle, we can generalize leave-one-out CV to leave-d-out CV. Leave a set $\mathcal{C}$ comprising d observations out (set them aside as test data) and use the remaining $n - d$ data points for training the statistical model or algorithm:

$$\hat{\theta}_{n-d}^{(-\mathcal{C})}, \quad \text{for all possible subsets } \mathcal{C}_k, \ k = 1, 2, \ldots, \tbinom{n}{d}.$$

We can then evaluate this estimate on observations from the test set $\mathcal{C}_i$ (the test sample), for every i. The analogue of (4.2), for regression with $\hat{m}(\cdot)$, is then

$$\binom{n}{d}^{-1} \sum_{k=1}^{\binom{n}{d}} d^{-1} \sum_{i \in \mathcal{C}_k} \rho\left(Y_i, \hat{m}_{n-d}^{(-\mathcal{C}_k)}(X_i)\right) \tag{4.3}$$

which is also an estimate of the test set error, or generalization error in (4.1).

The computational burden becomes immense if $d \geq 3$. A computational short-cut is then given by randomization: instead of considering *all* subsets (test sets), we draw B **random** test subsets

$$\mathcal{C}_1^*, \ldots, \mathcal{C}_B^* \text{ i.i.d. } \sim \text{Uniform}(\{1, \ldots, \binom{n}{d}\}),$$

where the Uniform distribution assigns probability $\binom{n}{d}^{-1}$ to every possible subset of size d. Such a Uniform distribution, or such a random subset $\mathcal{C}^*$, is easily constructed by **sampling without replacement**:

Draw d times randomly without replacement from $\{1, \ldots, n\}$, yielding a subset $\mathcal{C}^*$.

The random approximation for (4.3) is then

$$B^{-1} \sum_{k=1}^{B} d^{-1} \sum_{i \in \mathcal{C}_k^*} \rho\left(Y_i, \hat{m}_{n-d}^{(-\mathcal{C}_k^*)}(X_i)\right). \tag{4.4}$$

For $B = \infty$, the two expressions in (4.4) and (4.3) coincide (note that we only would need a finite, maybe huge, amount of computation for evaluation of (4.3)).

In practice, we typically choose $d = [\gamma n]$ with $\gamma \approx 0.1$ (10% test data). For the number of random test and training sets, we choose $B \approx 50 - 500$, depending on the cost to compute $\hat{\theta}_{n-d}^{(-\mathcal{C})}$ for a training set of size $n - d$ (evaluation on the test set $\mathcal{C}$) is usually fast). Thus, in terms of computation, the stochastic version in (4.4) may be even faster than leave-one-out CV in (4.2) if $B < n$, and we can use the stochastic approximation also for leave-one-out CV when sample size n is large.

4.4 Properties of different CV-schemes

4.4.1 Leave-one-out CV

Leave-one-out CV, which is equal to n-fold CV, is approximately unbiased for the true prediction or generalization error: the only drawback in terms of bias is that we use training sample size $n - 1$ instead of the original n, causing for a slight bias. The variance of leave-one-out CV is typically high, because the n training sets are so similar to each other:

$$\text{Var}\left(n^{-1} \sum_{i=-1}^{n} \rho\left(Y_i, \hat{m}_{n-1}^{(-i)}(X_i)\right)\right) = n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} \text{Cov}\left(\rho(Y_i, \hat{m}_{n-1}^{(-i)}(X_i)), \rho(Y_j, \hat{m}_{n-1}^{(-j)}(X_j))\right).$$

Although (X_i, Y_i) is typically assumed to be independent from (X_j, Y_j), the covariances are substantial because of strong correlation of $\hat{m}_{n-1}^{(-i)}(\cdot)$ and $\hat{m}_{n-1}^{(-j)}(\cdot)$, and hence the double sum in the formula above can be quite large.

4.4.2 Leave-d-out CV

Heuristically, it is clear that leave-d-out CV, with $d > 1$, has higher bias than leave-one-out CV; because we use training samples of sizes $n - d$ instead of the original n, causing some bias. In terms of variance, we average over more, although highly correlated summands in (4.3), which can be shown to decrease variance in comparison to leave-one-out CV.

4.4.3 K-fold CV; stochastic approximations

K-fold CV has larger bias than leave-one-out CV; because the training sets are of smaller sample size than $n-1$ (in leave-one-out CV), and also smaller than the original sample size n. In terms of variance, it is not clear (sometimes wrongly stated in the literature) whether K-fold CV has smaller variance than leave-one-out CV.

The stochastic approximation is expected to have somewhat higher bias and variance than the computationally infeasible leave-d-out CV: it is difficult to assess how much we lose by using a finite B.

4.5 Computational shortcut for some linear fitting operators

Consider the special case of fitting a cubic smoothing spline or fitting a least squares parametric estimator: in both cases, we have a linear fitting operator $\mathcal{S}$,

$$(\hat{m}(x_1), \ldots, \hat{m}(x_n))^\intercal = \mathcal{S}\mathbf{Y}, \ \mathbf{Y} = (Y_1, \ldots, Y_n)^\intercal.$$

When focusing on the squared error loss function $\rho(y,x) = |y-x|^2$, there is a surprising result for representing the leave-one-out CV score in (4.2):

$$n^{-1}\sum_{i=1}^{n}\left(Y_i - \hat{m}_{n-1}^{(-i)}(X_i)\right)^2 \quad = \quad n^{-1}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{m}(X_i)}{1-\mathcal{S}_{ii}}\right)^2. \tag{4.5}$$

The interesting property is that we can compute the CV score by fitting the original estimator $\hat{m}(\cdot)$ **once on the full dataset**, without having to do it n times by holding one observation back as a test point. Moreover, by using efficient linear algebra implementations, the elements $\mathcal{S}_{ii}$ can be computed with $O(n)$ operations.

Historically, it was computationally more difficult to obtain all diagonal elements from the hat matrix $\mathcal{S}_{ii}$ $(i = 1, \ldots, n)$; and computing the trace$(\mathcal{S})$, which equals the sum of the eigenvalues of $\mathcal{S}$, has been easier. The generalized cross-validation was then proposed in the late 70's:

$$\mathrm{GCV} = \frac{n^{-1}\sum_{i=1}^{n}(Y_i - \hat{m}(X_i))^2}{(1-n^{-1}\mathrm{trace}(\mathcal{S}))^2}.$$

This again requires to compute $\hat{m}(\cdot)$ only once on the full dataset. Moreover, if all the diagonal elements $\mathcal{S}_{ii}$ would be equal (which is typically not the case), GCV would coincide with the formula in (4.5). As outlined already, GCV has been motivated by computational considerations which are nowadays not very relevant anymore. However, the statistical properties of GCV can also be as good (and sometimes better or worse) than the ones from leave-one-out CV.