Chapter 2

Weber Jakob

November 18, 2020

Contents

1 Fundamentals **2**
1.1 Linear Models 2
1.1.1 Parameter Estimation 4
1.1.1.1 The Method of Least Squares 4
1.1.1.2 Maximum Likelihood Estimation 5
1.1.2 Estimation of the Variance σ^2 5
1.1.2.1 Maximum Likelihood Estimation 5
1.1.2.2 Restricted Maximum Likelihood Estimation . . 6
1.1.3 The Hat Matrix 6
1.2 Model Selection 6
1.2.1 Model Assessment Criteria 7
1.2.1.1 Sum of Expected Squared Prediction Error . . . 7
1.2.1.2 Corrected Coefficient of Determination R^2_{corr} . . 9
1.2.1.3 Corrected Coefficient of Determination $R^2_{McFadden}$ 10
1.2.1.4 Mallow's Cp 10
1.2.1.5 Akaike Information Criterion 10
1.2.1.6 Bayesian Information Criteria 11
1.2.1.7 Cross-validation 11
1.2.2 Subset Selection Methods 12
1.2.3 Regularization 12
1.3 Splines 13
1.3.1 B-Splines 13
1.3.1.1 Tensor-Product B-Splines 17
1.3.1.2 Additive Regression 17
1.3.2 P-Splines 19

Chapter 1

Fundamentals

This chapter summarizes the fundamentals of regression. Excellent overviews can be found in the textbooks [1], [2], [3]. The shown fundamentals are strongly aligned with the presentation given in [2]. Section 1.1 gives an overview of the model assumptions used throughout this work, Furthermore, Section 1.2 outlines methods to evaluate and compare different models again each other in terms of complexity and accuracy. Section 1.3 is devoted to the spline definitions.

1.1 Linear Models

Given the data set $D = \{(x_1^{(i)}, \dots, x_q^{(i)}, y^{(i)}),\ i = 1, 2, \dots, n\}$, we aim to model the relation between the inputs $x_1, \dots, x_q$ and the output y with a function $f(x_1, \dots, x_q)$. Since this relationship is not exact, there will be a random part ϵ. It is typically assumed that this part is additive and thus

$$y = f(x_1, \dots, x_q) + \epsilon. \tag{1.1}$$

We would like to estimate the unknown function f. For this, some assumptions on the model structure are made:

1. *The unknown function f is a linear combination of the inputs*

   The function $f(x_1, \dots, x_q)$ is modeled as a linear combination of inputs, i.e. in the form

   $$f(x_1, \dots, x_q) = \beta_0 + \beta_1 x_1 + \dots + \beta_q x_q, \tag{1.2}$$

   with unknown parameters $\beta_0, \dots, \beta_q$. Note that the model (1.2) is linear in its parameters as well as in its inputs. The parameter β_0 is called intercept or bias in the machine learning community, see [4]. We introduce the input vectors $\mathbf{x}^{\mathrm{T}} = [1, x_1, \dots, x_q] \in \mathbb{R}^{1 \times q+1}$ and the parameter vector $\boldsymbol{\beta}^{\mathrm{T}} = [\beta_0, \beta_1, \dots, \beta_q] \in \mathbb{R}^{1 \times q+1}$ to obtain

   $$y(x_1, \dots, x_q) = y(\mathbf{x}^{\mathrm{T}}) = \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}. \tag{1.3}$$

2. *Additive errors*

   An additional assumption of linear models is additivity of errors, which means that
$$y = \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} + \epsilon. \tag{1.4}$$
   This is reasonable for many practical applications, even though it appears quite restrictive.

To estimate the unknown parameters or coefficients $\boldsymbol{\beta}$, we define the output vector $\mathbf{y}^{\mathrm{T}} = [y^{(1)}, \ldots, y^{(n)}] \in \mathbb{R}^{1\times n}$ and data error vector $\boldsymbol{\epsilon}^{\mathrm{T}} = [\epsilon^{(i)}, \ldots, \epsilon^{(n)}] \in \mathbb{R}^{1\times n}$ as well as the design matrix
$$\mathbf{X} = \begin{bmatrix} 1 & x_1^{(1)} & \ldots & x_q^{(1)} \\ \vdots & & & \vdots \\ 1 & x_1^{(n)} & \ldots & x_q^{(n)} \end{bmatrix} \in \mathbb{R}^{n\times q+1} \tag{1.5}$$
and generate n equations like (1.4), which can be combined to
$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{1.6}$$
We assume that the design matrix $\mathbf{X}$ has full column rank, i.e. $\mathrm{rank}(\mathbf{X}) = q + 1 = p$, implying linear independence of the columns of $\mathbf{X}$, which is necessary to obtain a unique estimator for the regression coefficients $\boldsymbol{\beta}$, see [2].

Another necessary requirement is that the number of data points n is larger or equal to the number of regression parameters p, which is equivalent to the statement that the linear system in (1.6) is not underdetermined.

In addition to the assumptions on the unknown function f, the necessary assumptions on the error term ϵ_i are the following [2]:

1. *Expectation of the error*

   The errors have a mean value of zero, i.e. $\mathrm{E}(\epsilon^{(i)}) = 0$

2. *Variances and correlation structure of the errors*

   We assume constant error variance with $\mathrm{Var}(\epsilon^{(i)}) = \sigma^2$. This property is called homoscedasticity. Additionally, we assume that the errors are uncorrelated, which means $\mathrm{Cov}(\epsilon^{(i)}, \epsilon^{(j)}) = 0$ for $i \neq j$. The combination of these assumptions lead to the covariance matrix $\mathrm{Cov}(\boldsymbol{\epsilon}) = \mathrm{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}) = \sigma^2\mathbf{I}$.

3. *Assumptions on the input and design matrix*

   We have to distinguish two cases where the inputs are deterministic or stochastic. In most cases, the inputs and the output are stochastic and hence all model assumptions are conditioned on the design matrix (1.5). This means that the input $\mathbf{x}^{(i)}$ and the errors ϵ_i are not stochastically independent. For notational simplicity, we usually suppress the dependence on the design matrix.

4. *Gaussian errors*

   The errors follow at least approximately a normal distribution. Together with assumptions 1 and 2, we obtain that $\epsilon^{(i)} = \mathcal{N}(0, \sigma^2)$.

Summarizing, we have the following model assumptions:

$$\mathrm{E}(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta} \tag{1.7}$$

$$\mathrm{Cov}(\mathbf{y}) = \sigma^2\mathbf{I}, \tag{1.8}$$

yielding

$$\mathbf{y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}). \tag{1.9}$$

A linear model with multiple inputs can therefore be interpreted as a multivariate normal distribution with its mean vector given by $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$ and its covariance matrix given by $\sigma^2\mathbf{I}$, i.e. to specify the linear model given in (1.9), we need to estimate the regression coefficients $\boldsymbol{\beta}$ and the variance σ^2.

1.1.1 Parameter Estimation

The linear model given in (1.9) features the unknown parameters $\boldsymbol{\beta}$ and σ, which need to be estimated using the given data. In the following, the estimators $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}$ are introduced, and their statistical properties are derived. The two methods to estimate the regression parameters in the context of linear models are the method of Least Squares (LS) and the method of Maximum Likelihood (ML).

1.1.1.1 The Method of Least Squares

The unknown regression parameters $\boldsymbol{\beta} \in \mathbb{R}^p$ are estimated by minimizing the sum of squared error

$$\mathrm{LS}(\mathbf{y}, \boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = \sum_{i=1}^{n} \epsilon_i^2 = \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon}, \tag{1.10}$$

with respect to $\boldsymbol{\beta}$ [3]. Rewriting (1.10) leads to the least squares criterion

$$\begin{aligned} \mathrm{LS}(\mathbf{y}, \boldsymbol{\beta}) &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta}. \end{aligned} \tag{1.11}$$

The first-order necessary condition for optimality, cf. [5], reads as

$$\frac{\partial \mathrm{LS}(\mathbf{y}, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^{\mathrm{T}}\mathbf{y} + 2\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} = 0. \tag{1.12}$$

The second-order condition requires the Hessian to be positive-definite, i.e.

$$\frac{\partial^2 \mathrm{LS}(\mathbf{y}, \boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathrm{T}}} = 2\mathbf{X}^{\mathrm{T}}\mathbf{X} > 0. \tag{1.13}$$

Since the design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ is assumed to have full rank, the matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is positive definite. The least squares estimate $\hat{\boldsymbol{\beta}}_{LS}$ is hence obtained, see (1.12), by solving the so-called *normal equations*

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{1.14}$$

Since $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is positive definite, the *normal equations* (1.14) feature a unique solution given by the least squares estimator

$$\hat{\boldsymbol{\beta}}_{LS} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{1.15}$$

1.1.1.2 Maximum Likelihood Estimation

Under the normality assumption, the likelihood is defined as [1]

$$\mathcal{L}(\boldsymbol{\beta}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\Big(- \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\Big). \tag{1.16}$$

The log-likelihood is then given by

$$l(\boldsymbol{\beta}, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \tag{1.17}$$

Thus, maximizing the log-likelihood $l(\boldsymbol{\beta}, \sigma^2)$ with respect to $\boldsymbol{\beta}$ is equivalent to minimizing the least squares criterion given in (1.10). The maximum likelihood estimator $\hat{\boldsymbol{\beta}}_{ML}$ is therefore equivalent to the least squares estimator $\hat{\boldsymbol{\beta}}_{LS}$ in (1.15).

1.1.2 Estimation of the Variance σ^2

The estimation of the variance σ^2 is necessary for the construction of confidence intervals of the regression parameters and for the construction of prediction intervals. It is further used in all kinds of statistical tests as well as in model selection approaches and model assessment criteria [6].

1.1.2.1 Maximum Likelihood Estimation

The first-order necessary condition for optimality results in this case in

$$\frac{\partial l(\boldsymbol{\beta}, \sigma^2)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0. \tag{1.18}$$

Substituting the maximum likelihood estimator $\hat{\boldsymbol{\beta}}_{LS}$, given in (1.15), for $\boldsymbol{\beta}$ results in the maximum likelihood estimator

$$\hat{\sigma}^2_{ML} = \frac{(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{LS})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{LS})}{n} = \frac{\hat{\boldsymbol{\epsilon}}^{\mathrm{T}}\hat{\boldsymbol{\epsilon}}}{n}. \tag{1.19}$$

This estimator for σ^2 is rarely used since it is biased, i.e. $\mathrm{E}(\hat{\sigma}^2_{ML}) \neq \sigma^2$, see [2].

1.1.2.2 Restricted Maximum Likelihood Estimation

The mean value of the sum of squared residuals is $\mathrm{E}(\hat{\boldsymbol{\epsilon}}^{\mathrm{T}}\hat{\boldsymbol{\epsilon}}) = (n-p)\sigma^2$. Hence, a less biased estimator $\hat{\sigma}^2$ for σ^2 is given by

$$\hat{\sigma}^2_{REML} = \frac{1}{n-q}\hat{\boldsymbol{\epsilon}}^{\mathrm{T}}\hat{\boldsymbol{\epsilon}}. \tag{1.20}$$

The restricted maximum likelihood estimator for the variance is in general less biased [2]. Therefore, it is the commonly used estimator for the variance σ^2.

1.1.3 The Hat Matrix

Using the least squares estimator (1.15), we can estimate the mean of $\mathbf{y}$ by

$$\widehat{\mathrm{E}(\mathbf{y})} = \hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{LS} \tag{1.21}$$

Substituting (1.15) results in

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{H}\mathbf{y}, \tag{1.22}$$

with the matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$, which is called *hat matrix*. Using the *hat matrix*, we can express the residuals $\hat{\epsilon}^{(i)} = y^{(i)} - \hat{y}^{(i)}$ in matrix notation as

$$\hat{\boldsymbol{\epsilon}} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}. \tag{1.23}$$

The *hat matrix* $\mathbf{H}$ has the following useful properties:

- $\mathbf{H}$ is symmetric.
- $\mathbf{H}$ is idempotent, i.e. $\mathbf{H}^2 = \mathbf{H}$.
- The rank of $\mathbf{H}$ is equal to its trace.
- $\frac{1}{n} \leq h_{ii} \leq 1$, if all data points are different, i.e. $x^{(i)} \neq x^{(j)}$ for $i \neq j$. Here, h_{ii} are the diagonal elements of $\mathbf{H}$.
- The matrix $(\mathbf{I}-\mathbf{H})$ is also idempotent and symmetric, with $\mathrm{rank}(\mathbf{I}-\mathbf{H}) = n-p$.

The hat matrix is used in model selection techniques like cross-validation as well as in outlier detection and in diagnostic plots for linear models.

1.2 Model Selection

Linear models are widely exploited for regression problems on large data sets ($n \gg 0$), because the solution of the *normal equations* (1.14) can be computed efficiently even for large n. If these data sets also contain a large number of input variables ($q \gg 0$), the situation becomes more complicated since possible interaction effects or correlation of input variables may occur. This interaction terms limits the, otherwise perfect, interpretability of the linear model.

Therefore, we need techniques and criteria to select the *best possible model* out of the variety of different models for a given data set. Model assessment criteria, see Section 1.2.1, are used to compare different models while subset selection techniques, see Section 1.2.2, give an algorithmic approach to model generation. Further, we can influence the estimated coefficients $\boldsymbol{\beta}$ directly via regularization, see Section 1.2.3.

1.2.1 Model Assessment Criteria

One way of comparing various models, i.e models using different sets of inputs, is the use of model assessment criteria. Generally, model assessment criteria can be split in two components. The first one measures the goodness of fit, e.g. using the sum of squared errors, while the second measures the complexity of the model. Most model assessment criteria are based on the sum of the expected squared prediction error (SPSE), which is also known as *generalization error*. Therefore, the derivation of the SPSE is given next.

1.2.1.1 Sum of Expected Squared Prediction Error

Given independent observations $y_i,\ i = 1, 2, \dots, n$ and a subset of inputs $\{x_0 = 1, x_1, \dots, x_q\}$, we want to measure the prediction quality. The specific models are defined by numbers $M \subset \{0, 1, \dots, q\}$ of used inputs with corresponding design matrix $\mathbf{X}_M$. Moreover, $|M|$ is the cardinal number of M, i.e. the number of inputs included in the model. The least squares estimator for $\boldsymbol{\beta}$, cf. (1.15), is then given by

$$\hat{\boldsymbol{\beta}}_M = (\mathbf{X}_M^{\mathrm{T}}\mathbf{X}_M)^{-1}\mathbf{X}_M^{\mathrm{T}}\mathbf{y}.$$

The data $\mathbf{y}$ can be interpreted as random variable. We can then define an estimator $\hat{\mathbf{y}}_M$ for the vector $\boldsymbol{\mu}$ of expectations $\mu^{(i)} = \mathrm{E}(y^{(i)})$ by

$$\hat{\mathbf{y}}_M = \mathbf{X}_M\hat{\boldsymbol{\beta}}_M. \tag{1.24}$$

Moreover, it is easy to show that the following properties hold true:

(i) $\mathrm{E}(\hat{\mathbf{y}}_M) = \mathbf{X}_M(\mathbf{X}_M^{\mathrm{T}}\mathbf{X}_M)^{-1}\mathbf{X}_M^{\mathrm{T}}\mathrm{E}(\mathbf{y})$

(ii) $\mathrm{Cov}(\hat{\mathbf{y}}_M) = \sigma^2\mathbf{X}_M(\mathbf{X}_M^{\mathrm{T}}\mathbf{X}_M)^{-1}\mathbf{X}_M^{\mathrm{T}}$

(iii) $\sum_{i=1}^n \mathrm{Var}(\hat{y}_M^{(i)}) = \sigma^2\mathrm{tr}(\mathbf{X}_M(\mathbf{X}_M^{\mathrm{T}}\mathbf{X}_M)^{-1}\mathbf{X}_M^{\mathrm{T}}) = \sigma^2|M|$

(iv) Sum of Mean Squared Error

$$\begin{aligned}\mathrm{SMSE} &= \sum_{i=1}^n \mathrm{E}(\hat{y}_M^{(i)} - \mu_M^{(i)})^2\\ &= \sum_{i=1}^n \mathrm{E}\Big(\big(\hat{y}_M^{(i)} - \mathrm{E}(\hat{y}_M^{(i)})\big) + \big(\mathrm{E}(\hat{y}_M^{(i)}) - \mu_M^{(i)}\big)\Big)^2 \\ &= |M|\sigma^2 + \sum_{i=1}^n \big(\mathrm{E}(\hat{y}_M^{(i)}) - \mu_M^{(i)}\big)^2.\end{aligned} \tag{1.25}$$

Note that the estimator (1.24) can be regarded as a prediction of future observations

$$y^{(n+i)} = \mu^{(i)} + \epsilon^{(n+i)} \tag{1.26}$$

for new input data $\{x_1^{(i)}, \ldots, x_q^{(i)}\}$. Thus, we can derive the SPSE as

$$\begin{aligned}
\text{SPSE} &= \sum_{i=1}^{n} \mathrm{E}(y^{(n+i)} - \hat{y}_M^{(i)})^2 \\
&= \sum_{i=1}^{n} \mathrm{E}\big((y^{(n+i)} - \mu_M^{(i)}) - (\hat{y}_M^{(i)} - \mu_M^{(i)})\big)^2 \\
&= \sum_{i=1}^{n} \mathrm{E}(y^{(n+i)} - \mu_M^{(i)})^2 + 2\mathrm{E}\big((y^{(n+i)} - \mu^{(i)})(\hat{y}_M^{(i)} - \mu_M^{(i)})\big) + \mathrm{E}(\hat{y}^{(i)} - \mu_M^{(i)})^2 \\
&= \sum_{i=1}^{n} \mathrm{E}(y^{(n+i)} - \mu_M^{(i)})^2 + \sum_{i=1}^{n} \big(\mathrm{E}(\hat{y}_M^{(i)}) - \mu_M^{(i)}\big)^2 \\
&= n\sigma^2 + \text{SMSE} \\
&= n\sigma^2 + |M|\sigma^2 + \sum_{i=1}^{n} \big(\mathrm{E}(\hat{y}^{(i)}) - \mu_M^{(i)}\big)^2 .
\end{aligned} \tag{1.27}$$

The SPSE can be split into three parts:

1. *Irreducible Prediction Error Term*: $n\sigma^2$
   This term cannot be reduced through model selection techniques since it only contains the number of data points n and the variance σ^2.

2. *Variance Error Term*: $|M|\sigma^2$
   The second term contains the number of used variables $|M|$ as well as the variance σ^2. It can therefore be reduced by reducing the model complexity, i.e. by using a smaller number of inputs.

3. *Squared Bias Error Term*: $\sum_{i=1}^{n} \big(\mathrm{E}(\hat{y}^{(i)} - \mu_M^{(i)})\big)^2$
   The last term can be interpreted as bias. It can be reduced by increasing the model complexity, i.e. by using additional inputs.

The SPSE acts as an example of the bias-variance trade-off, which is characteristic for all statistical models. It states that by increasing the model complexity, the bias is reduced but instead the variance is increased. On the other hand, by decreasing model complexity, the variance of the model is reduced, but the bias is increased, see Figure 1.1 [4].

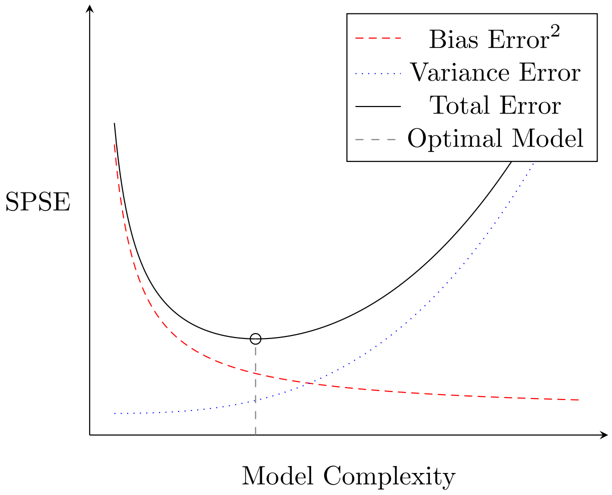

Figure 1.1: Bias-variance decomposition

In practice, the true value for the SPSE is not accessible since $\mu^{(i)}$ and σ^2 are unknown. Therefore, we need to estimate the SPSE. This can be done by using one of the following two strategies:

1. *Estimate SPSE using new and independent data*
   If new observations are available, the SPSE can be estimated by

$$\widehat{\text{SPSE}} = \sum_{i=1}^{n} (y^{(n+i)} - \hat{y}_M^{(i)})^2. \tag{1.28}$$

   These new observations can also be some held-out validation data from a train-validation split of the given data.

2. *Estimate SPSE using existing data*
   When using existing data, the estimate for the SPSE is given by the squared error and an additional term depending on the estimated variance and the model complexity. The estimate is thus given by

$$\widehat{\text{SPSE}} = \sum_{i=1}^{n} (y^{(i)} - \hat{y}_M^{(i)})^2 + 2|M|\hat{\sigma}^2. \tag{1.29}$$

Typically used model assessment criteria follow the basic idea of the SPSE, see [2].

1.2.1.2 Corrected Coefficient of Determination R^2_{corr}

The corrected coefficient of determination R^2_{corr} is an improvement over the coefficient of determination R^2, which is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y^{(i)} - \hat{y}_M^{(i)})^2}{\sum_{i=1}^{n} (y^{(i)} - \bar{y})^2}, \tag{1.30}$$

where $\bar{y}$ is defined as the mean value of $\mathbf{y}$. The major drawback of R^2 is that it will never decrease when further inputs are included in the model, e.g. the R^2 of a model using $\{x_1, x_2, x_3\}$ is always larger or equal the R^2 of a model using $\{x_1, x_2\}$, even if the variable does not enhance the prediction quality.

The corrected coefficient of determination R^2_{corr} reduces this problem by an correction term depending on the number of parameters and is given by

$$R^2_{corr} = 1 - \frac{n-1}{n-p}(1 - R^2). \tag{1.31}$$

The corrected coefficient of determination is a standard output parameter in many statistical programs and may be used to compare even models with different number of used variables [2].

1.2.1.3 Corrected Coefficient of Determination $R^2_{McFadden}$

The corrected coefficient of determination after McFadden is defined as

$$R^2_{McFadden} = 1 - \frac{\ln L_M - M}{\ln L_0} \tag{1.32}$$

using the likelihood of the model M given by L_M and the likelihood of the zero model L_0. A standard zero model is given by the mean value $\mathrm{E}(y^{(i)})$. Higher values of $R^2_{McFadden}$ correspond to better fits.

1.2.1.4 Mallow's Cp

Mallow's complexity parameter is based directly on the ideas specified for the estimation of the SPSE and is given by

$$\mathrm{C}_p = \frac{\sum_{i=1}^{n}(y^{(i)} - \hat{y}_M^{(i)})^2}{\hat{\sigma}^2} - n + 2|M|. \tag{1.33}$$

A lower value of Mallow's C_p corresponds to a better model fit [2].

1.2.1.5 Akaike Information Criterion

The AIC is among the most used model assessment criteria and defined by

$$\mathrm{AIC} = -2l(\hat{\boldsymbol{\beta}}_{ML}, \hat{\sigma}^2_{ML}) + 2(|M| + 1), \tag{1.34}$$

where $l(\hat{\boldsymbol{\beta}}_{ML}, \hat{\sigma}^2_{ML})$ is the value of the log-likelihood (1.17) at its maximum, i.e. at $\hat{\boldsymbol{\beta}}_{ML}$ and $\hat{\sigma}_{ML}$. It is worth noting that the total number of parameters is $|M| + 1$ because the variance is also counted as parameter. The log-likelihood for a linear model assuming Gaussian errors is given by, cf. (1.17),

$$-2l(\hat{\boldsymbol{\beta}}_{ML}, \hat{\sigma}^2_{ML}) = n \log(\hat{\sigma}^2_{ML}) + n. \tag{1.35}$$

Therefore, neglecting the constant n, the AIC evaluates to

$$\mathrm{AIC} = n \log(\hat{\sigma}^2_{ML}) + 2(|M| + 1). \tag{1.36}$$

A lower value of the AIC means a to a better model fit [2].

1.2.1.6 Bayesian Information Criteria

The BIC is similar to the AIC, but it penalizes more complex models much harder than the AIC. In its general form, it is given as

$$\mathrm{BIC} = -2l(\hat{\boldsymbol{\beta}}_{ML}, \hat{\sigma}^2_{ML}) + \log(n)(|M| + 1). \tag{1.37}$$

Again, assuming Gaussian errors for a linear model and neglecting the constant term n, the BIC evaluates to

$$\mathrm{BIC} = n \log(\hat{\sigma}^2_{ML}) + \log(n)(|M| + 1). \tag{1.38}$$

A lower value of the BIC correspond to a better model fit [2].

1.2.1.7 Cross-validation

The basic idea of cross-validations is to split the given data set into a training set to estimate the parameters and a validation set to assess the prediction quality. A special case of cross-validation is the "leave-one-out" cross-validation, where all but one data point are used for training and the model is then evaluated on this held-out data point. This seems to be quite expensive, since one needs to estimate one model per data point. However, in the context of linear models, it can be shown that the cross-validation score can be computed using one model trained on all data $\mathbf{y}$ and the *hat matrix* $\mathbf{H}_M = \mathbf{X}_M(\mathbf{X}_M^{\mathrm{T}}\mathbf{X}_M)^{-1}\mathbf{X}_M^{\mathrm{T}}$. The cross-validation score is then given by

$$\mathrm{CV} = \frac{1}{n}\sum_{i=1}^{n}\Big(\frac{y^{(i)} - \hat{y}^{(i)}_M}{1 - h_{ii,M}}\Big)^2. \tag{1.39}$$

where $h_{ii,M}$ denote the diagonal elements of the *hat matrix* and $\hat{y}^{(i)}_M$ is defined as the prediction of the model for the input $\{x_1^{(i)}, \ldots, x_q^{(i)}\}$. A lower cross-validation score corresponds to a better model fit [7].

An approximation to the cross-validation score is given by the so-called generalized cross-validation score. It is mainly used in the context of non-parametric regression or when the hat matrix $\mathbf{H}_M$ is numerically expensive to compute. In the GCV score, the diagonal elements of the hat matrix $h_{ii,M}$ are replaced by the mean of the trace of $\mathbf{H}_M$. The GCV score is then given by

$$\mathrm{GCV} = \frac{1}{n}\sum_{i=1}^{n}\Big(\frac{y^{(i)} - \hat{y}^{(i)}_M}{1 - \mathrm{trace}(\mathbf{H}_M)/n}\Big)^2. \tag{1.40}$$

The numerical advantage comes from the fact that the trace of a product of matrices is invariant to cyclical permutations, i.e.

$$\mathrm{trace}(\mathbf{H}_M) = \mathrm{trace}(\mathbf{X}_M(\mathbf{X}_M^{\mathrm{T}}\mathbf{X}_M)^{-1}\mathbf{X}_M^{\mathrm{T}}) = \mathrm{trace}(\mathbf{X}_M^{\mathrm{T}}\mathbf{X}_M(\mathbf{X}_M^{\mathrm{T}}\mathbf{X}_M)^{-1}). \tag{1.41}$$

The trace can therefore be computed from the product of two matrices of shape $p \times p$ [2].

1.2.2 Subset Selection Methods

To make use of the various model assessment criteria, some algorithmic approach to model selection needs to be given. The most commonly used approaches are forward, backward and stepwise selction [2].

In forward selection, start with a candidate model, which includes a small number of variables. In each iteration of forward selection, an additional variable is added to the candidate model. The added variable is the one with leads to the largest reduction of a predefined model assessment criteria. The algorithm stops, if no further reduction is achieved.

In backward selection, start with a candidate model, which includes all variables. In each iteration of backward selection, we eliminate the variable from the model which provides the largest reduction of a predefined model assessment criteria. The algorithm stops, if no further reduction is possible.

In step-wise selection, forward and backward selection are combined to enable the inclusion and deletion of a variable in every operation. The algorithm stops, if no further reduction is possible.

1.2.3 Regularization

Model selection can also be achieved using regularization techniques by directly influencing the parameters $\boldsymbol{\beta}$, which need to be estimated given a data set. In general, regularization restricts the parameter space by adding some penalty term depending on the complexity of the model to the least squares objective function according to (1.10). This leads to the penalized least squares (PLS) criterion

$$\mathrm{PLS}(\mathbf{y}, \boldsymbol{\beta}; \lambda) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \cdot \mathrm{pen}(\boldsymbol{\beta}), \tag{1.42}$$

where λ is the so-called smoothing parameter and $\mathrm{pen}(\boldsymbol{\beta})$ is the penalty term describing the regularization technique.

In Ridge regression, the penalty term in the penalized least squares criterion in (1.42) is given by the squared weighted L_2-norm of the parameter vector $\boldsymbol{\beta}$, i.e. $\mathrm{pen}(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_{\mathbf{K}}^2 = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{K}\boldsymbol{\beta}$ with penalty matrix $\mathbf{K} \in \mathbb{R}^{p \times p}$. The closed form solution reads as

$$\hat{\boldsymbol{\beta}}_{PLS} = \arg\min_{\boldsymbol{\beta}} \big(\mathrm{PLS}(\mathbf{y}, \boldsymbol{\beta}; \lambda)\big) = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{K})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}, \tag{1.43}$$

The additional penalty term in Ridge regression leads to smaller parameter estimates $\hat{\boldsymbol{\beta}}_{PLS}$ compared to the unpenalized estimate $\hat{\boldsymbol{\beta}}_{LS}$. For large values of the smoothing parameter λ, the parameter estimates will converge towards, but never reach, zero.

Ridge regression is commonly used when the input dimension q is high, i.e. the number of parameters β_i is large, and also known as Tikhonov regularization [8]. Note that it is also possible to use a nonlinear penalty matrix $\mathbf{K}(\boldsymbol{\beta})$ resulting in

$$\mathrm{pen}(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_{\mathbf{K}(\boldsymbol{\beta})}^2 = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{K}(\boldsymbol{\beta})\boldsymbol{\beta}. \tag{1.44}$$

However, the resulting penalized least squares problem has no closed form solution and must be solved by an iterative approach. We start with an initial guess $\boldsymbol{\beta}^{[0]}$ and iterate for $k = 0, 1, 2, \ldots$ the iteration

$$\boldsymbol{\beta}^{[k+1]} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{K}(\boldsymbol{\beta}^{[k]})\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{1.45}$$

until $|\boldsymbol{\beta}^{[k+1]} - \boldsymbol{\beta}^{[k]}| \leq \mathrm{Tol}$ with Tol being some given tolerance.

1.3 Splines

A spline is a piecewise polynomial defined on a sequence of knots. This definition is quite general. Therefore, a large variety of splines exists, ranging from regression splines [9], over B-splines [10] to natural cubic splines and many more. We will focus on the definition of B-splines in Section 1.3.1, tensor-product B-splines as the multi-dimensional expansion of B-splines in Section 1.3.1.1, and P-splines in Section 1.3.2 [2] [11].

1.3.1 B-Splines

We put the focus on the definition and use of B-splines $s(x)$, which are constructed using the d B-spline basis functions $B_j^l(x)$ of order l as

$$s(x) = \sum_{j=1}^{d} B_j^l(x)\beta_j \tag{1.46}$$

given the knot sequence

$$K = \{\kappa_{1-l}, \kappa_{1-l+1}, \ldots, \kappa_{d+1}\}. \tag{1.47}$$

The B-spline basis function $B_j^l(x)$ of degree l is defined by means of the Cox-de Boor recursion formula as

$$B_j^0(x) = \begin{cases} 1 & \text{for} \quad \kappa_j \leq x < \kappa_{j+1} \\ 0 & \text{otherwise} \end{cases} \tag{1.48}$$

$$B_j^l(x) = \frac{x - \kappa_{j-l}}{\kappa_j - \kappa_{j-l}} B_{j-1}^{l-1}(x) + \frac{\kappa_{j+1} - x}{\kappa_{j+1} - \kappa_{j+1-l}} B_j^{l-1}(x) \tag{1.49}$$

using the knot sequence (1.47). Hence it is composed of $(l+1)$-polynomial pieces of degree l [2]. An example of a B-spline basis function of order $l = 0, 1, 2, 3$ is given in Figure 1.2.

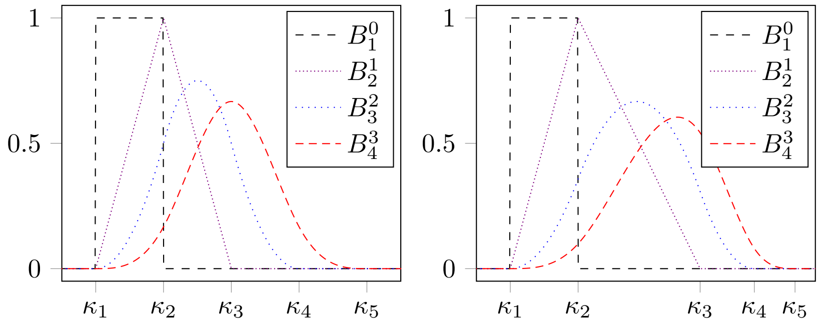

Figure 1.2: B-spline basis function of order $l = 0, 1, 2, 3$ for equidistant (left) and non-equidistant (right) knots

The left chart shows the B-spline basis functions based on an equidistant sequence of knots. The B-spline basis function B_1^0 is the zero function, except for $x \in [\kappa_1, \kappa_2]$ where it is equal to 1, see (1.48). The B-spline basis function B_2^1 is the well known *hat function*, being zero except for $x \in [\kappa_1, \kappa_3]$. It consists of two linear pieces, one defined from κ_1 to κ_2, the other from κ_2 to κ_3. This can be seen by expanding the recursive definition (1.49) as

$$B_2^1(x) = \frac{x - \kappa_1}{\kappa_2 - \kappa_1} B_1^0(x) + \frac{\kappa_3 - x}{\kappa_3 - \kappa_2} B_2^0(x). \tag{1.50}$$

Everywhere else, B_2^1 is equal to zero. At the joining points, the values of the linear pieces are equal. The B-spline basis function B_3^2 consists of three quadratic pieces, joining at the knots κ_2 and κ_3. Expanding the recursive definition, using equidistant knots with the knot spacing h, shows this as

$$\begin{aligned} B_3^2(x) &= \frac{x - \kappa_1}{\kappa_3 - \kappa_1} B_2^1(x) + \frac{\kappa_4 - x}{\kappa_4 - \kappa_2} B_3^1(x) \\ &= \dots \\ &= \frac{1}{2h^2} \left[(x - \kappa_1)^2 B_1^0(x) + [(x - \kappa_1)(\kappa_3 - x) + (\kappa_4 - x)(x - \kappa_2)] B_2^0(x) + (\kappa_4 - x)^2 B_3^0(x) \right]. \end{aligned} \tag{1.51}$$

At κ_2 and κ_3, the values of the quadratic pieces, as well as their first derivatives are equal. Finally, the B-spline basis function B_4^3 consists of 4 cubic pieces with the joining points at κ_2, κ_3 and κ_4 at which respective cubic polynomials possess equal values as well as equal first and second-order derivatives.

The right part of Figure 1.2 shows the B-spline basis functions of the same order $l = 0, 1, 2, 3$ defined on a non-equidistant knot sequence. The basic properties of these are the same as for B-spline basis function based on equidistant knots. The shown locality, i.e. being nonzero only over a sequence of $l+2$ knots, is a very attractive feature leading to an enhanced numerical stability compared to other types of splines. Some general properties of a B-spline basis function of degree l are summarized in the following list:

- It consists of $l+1$ polynomial pieces of degree l, e.g. a cubic B-spline basis function ($l = 3$) consists of 4 cubic pieces.

- The pieces join at l inner knots.
- At these knots, the derivatives up to order $l-1$ are continuous.
- The B-spline basis function is positive on the domain spanned by $l+2$ knots, everywhere else it is zero, e.g. for $l=2$, a sequence of 4 knots is necessary.
- At every given x, only $l+1$ B-spline basis functions are nonzero.

Using the definition of B-spline basis functions, see (1.48) and (1.49), the first-order derivative of a B-spline basis function of degree l can be given as

$$\frac{\partial}{\partial x} B_j^l(x) = l\big[\frac{1}{\kappa_j - \kappa_{j-l}} B_{j-1}^{l-1}(x) - \frac{1}{\kappa_{j+1} - \kappa_{j+1-l}} B_j^{l-1}(x)\big] \tag{1.52}$$

using B-spline basis functions of degree $l-1$. Higher order derivatives are obtained by using lower order B-spline basis functions, see [10].

As shown in Figure 1.2, the knots can either be an equidistant sequence, which facilitates the construction and estimation of the coefficients, or a non-equidistant sequence. For equidistant knots, we split the domain $[a,b]$ into $m-1$ intervals, i.e

$$h = \frac{b-a}{m-1}, \tag{1.53}$$

where m is given by $m = d - l + 1$ and obtain the sequence

$$\kappa_j = a + h(j-1), \quad j = 1, \dots, m. \tag{1.54}$$

Non-equidistant knot placement can be obtained using quantile-based knots, i.e. by using the $(j-1)/(m-1)$-quantiles for $j = 1, \dots, m$ of the observed inputs $x^{(1)}, \dots, x^{(n)}$ as knots. Using this approach, more knots are placed in the areas where lots of data is present. The boundary knots, i.e. $\{\kappa_{1-l}, \dots, \kappa_0\}$ on the left side and $\{\kappa_{d-l+2}, \dots, \kappa_{d+1}\}$ on the right side, are usually set to be apart from each other by at least the minimal knot distance [2].

The collection of d B-spline basis functions of degree l over a sequence $K = \{\kappa_{1-l}, \kappa_{1-l+1}, \dots, \kappa_{d+1}\}$ knots is called B-spline basis. The basis is created such that it covers the domain $[a,b]$, i.e.

$$\sum_{j=1}^{d} B_j^l(x) = 1 \text{ for } x \in [a,b]. \tag{1.55}$$

A function $f(x)$ can then be represented by

$$f(x) = \sum_{j=1}^{d} B_j^l(x)\beta_j = \mathbf{b}^{\mathrm{T}}\boldsymbol{\beta}, \tag{1.56}$$

using the B-spline basis functions $B_j^l(x)$ of appropriate degree l and the parameter vector $\boldsymbol{\beta}^{\mathrm{T}} = [\beta_1, \dots, \beta_d] \in \mathbb{R}^{1\times d}$. The basis functions can be given in vector notation as $\mathbf{b}^{\mathrm{T}} = [B_1^l(x), \dots, B_d^l(x)] \in \mathbb{R}^{1\times d}$. Using the set of data $D = \{(x^{(i)}, y^{(i)}),\ i = 1, 2 \dots, n\}$, the B-spline basis matrix for d splines of degree l is given by the matrix $\mathbf{B}$ as

$$\mathbf{B} = \begin{bmatrix} B_1^l(x^{(1)}) & \dots & B_d^l(x^{(1)}) \\ \vdots & & \vdots \\ B_1^l(x^{(n)}) & \dots & B_d^l(x^{(n)}) \end{bmatrix} \in \mathbb{R}^{n\times d}. \tag{1.57}$$

The n equations (1.56) can then be arranged as a linear model in the form

$$\mathbf{y} = \mathbf{B}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{1.58}$$

Once the basis matrix in (1.57) is given, the parameters $\boldsymbol{\beta}$ can be estimated using the Least Squares algorithm given in Section 1.1.1.1 by optimizing the objective function

$$\mathrm{LS}(\mathbf{y}) = \|\mathbf{y} - \mathbf{B}\boldsymbol{\beta}\|_2^2. \tag{1.59}$$

Therefore, the estimation is computationally efficient and easy to implement since closed-form solutions exists. Further, the advanced theoretical framework of linear models can be applied to use model selection and regularization approaches as well as to calculate e.g confidence intervals for the regression coefficients and the prediction.

The derivative of the function $f(x)$ in (1.56) can be calculated by summing over all d basis functions and including the estimated parameters $\boldsymbol{\beta}$ into the B-spline basis function derivative (1.52) as

$$\frac{\partial f(x)}{\partial x} = \frac{\partial}{\partial x}\sum_{j=1}^{d} B_j^l(x)\beta_j = l\sum_{j=2}^{d} \frac{\Delta\beta_j}{\kappa_j - \kappa_{j-l}} B_{j-1}^{l-1}(x). \tag{1.60}$$

The second-order derivative of $f(x)$ can be given as

$$\frac{\partial^2 f(x)}{\partial x^2} = \frac{\partial^2}{\partial x^2}\sum_{j=1}^{d} B_j^l(x) = l^2\sum_{j=3}^{d} \frac{\Delta^2\beta_j}{(\kappa_j - \kappa_{j-l})(\kappa_{j+1} - \kappa_{j+1-l})} B_{j-2}^{l-2}(x). \tag{1.61}$$

Here, the finite difference operators $\Delta\beta_j = \beta_j - \beta_{j-1}$ and $\Delta_j^{\beta} = \Delta(\Delta\beta_j) = \beta_j - 2\beta_{j-1} + \beta_{j-2}$ are used. Therefore, by estimating the B-spline parameters $\boldsymbol{\beta}$, we also generate an estimate for the derivatives of the function $f(x)$.

B-splines of appropriate order $l > 2$ produce smooths curves, i.e. first and second order derivatives are continuous, where the smoothness is mostly determined by the number of splines used. By using a low number d, the curve will be quite smooth, but possess a large data error. When using a high number of splines d, the data error will be small but the variance of the curve will be large. This is an example of the bias-variance trade-off, a classical problem of regression and supervised learning, see Section 1.2.1.1 [10].

1.3.1.1 Tensor-Product B-Splines

Tensor-product B-splines can be seen as the multi-dimensional extension of B-splines. We examine an example for two input dimensions x_1 and x_2. Note that tensor-product B-splines can be constructed for arbitrary dimensions using the approach given below. Assume that we have B-spline bases B_1 and B_2 available using d_1 and d_2 basis functions for the respective dimension and degree $l = 3$. For readability, we may omit the degree l in the further description. The tensor-product B-spline basis is then constructed by considering all pairwise products of the two B-spline basis functions, i.e.

$$B_{j,r}(x_1, x_2) = B_j^l(x_1) B_r^l(x_2) \tag{1.62}$$

for $j = 1, 2, \dots, d_1$ and $r = 1, 2, \dots, d_2$. We then obtain the basis function representation for the tensor-product B-spline $t(x_1, x_2)$ as

$$t(x_1, x_2) = \sum_{j=1}^{d_1} \sum_{r=1}^{d_2} B_{j,r}(x_1, x_2)\beta_{j,r}. \tag{1.63}$$

We can therefore combine the individual basis functions to generated new basis functions for the tensor-product B-spline. For any set of data $D = \{(x_1^{(i)}, x_2^{(i)}, y^{(i)}),\ i = 1, 2, \dots, n\}$, the relationship between the basis matrix $\mathbf{X}$ of the tensor-product B-spline and the basis matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ is given as

$$\mathbf{X} = \mathbf{B}_1 \odot \mathbf{B}_2 \tag{1.64}$$

where $\odot$ indicates the use of the row-wise Kronecker product, see Appendix A, $\mathbf{X} \in \mathbb{R}^{n \times d_1 d_2}$ denotes the tensor-product B-spline basis matrix, $\mathbf{B}_1 \in \mathbb{R}^{n \times d_1}$ denotes the B-spline basis matrix for dimension x_1 and $\mathbf{B}_2 \in \mathbb{R}^{n \times d_2}$ denotes the B-spline basis matrix for dimension x_2 [2].

We can then model a two dimensional function using the data D similar to (1.58) as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\gamma} + \boldsymbol{\epsilon}, \tag{1.65}$$

with the tensor-product B-spline basis matrix $\mathbf{X} \in \mathbb{R}^{n \times d_1 d_2}$ and the parameter vector $\boldsymbol{\gamma}^{\mathrm{T}} = [\gamma_1, \dots, \gamma_{d_1 d_2}] \in \mathbb{R}^{1 \times d_1 d_2}$.

This approach can in theory be repeated for as many input dimensions as required. In practice, modeling more than two input dimensions using tensor-product B-splines becomes infeasible because of the exponential increase of basis functions and therefore parameters to estimate.

1.3.1.2 Additive Regression

To circumvent the latter problem, we now assume the restrictive structure of additive models, see [2], given by

$$f_{add} = f(x_1, \dots, x_q) = f_1(x_1) + \dots + f_q(x_q). \tag{1.66}$$

Hence, we use one function $f_i(x_i)$ per input dimension. For some given data $D = \{(x_1^{(i)}, \dots, x_q^{(i)}, y^{(i)}),\ i = 1, 2, \dots, n\}$, by using a B-spline $s_i(x_i)$ for each function $f_i(x_i)$ we obtain a linear model

$$f_i(\mathbf{x}_i) = \mathbf{X}_{s_i}\boldsymbol{\beta}_{s_i} \tag{1.67}$$

where $\mathbf{X}_{s_i} \in \mathbb{R}^{n \times d_i}$ is the B-spline basis matrix using d_i B-spline basis functions for $i = 1, 2, \dots, q$, $\mathbf{x}_i^{\mathrm{T}} = [x_i^{(1)}, \dots, x_i^{(n)}] \in \mathbb{R}^{1 \times n}$ is the data vector of input dimension i and $\boldsymbol{\beta}_{s_i} \in \mathbb{R}^{d_i \times 1}$ are the parameters to estimate. This leads to the model structure

$$\mathbf{y} = \mathbf{X}_{s_1}\boldsymbol{\beta}_{s_1} + \dots + \mathbf{X}_{s_q}\boldsymbol{\beta}_{s_q} + \boldsymbol{\epsilon}, \tag{1.68}$$

which can be written as linear model by concatenation of the B-spline basis matrices and parameter vectors as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \begin{bmatrix}\mathbf{X}_{s_1} : & \dots & : \mathbf{X}_{s_q}\end{bmatrix} \begin{bmatrix}\boldsymbol{\beta}_{s_1} \\ \vdots \\ \boldsymbol{\beta}_{s_q}\end{bmatrix} + \boldsymbol{\epsilon}, \tag{1.69}$$

with the matrix $\mathbf{X} \in \mathbb{R}^{n \times \sum_{i=1}^{q} d_i}$ and parameter vector $\boldsymbol{\beta} \in \mathbb{R}^{\sum_{i=1}^{q} d_i \times 1}$. The model (1.69) does not contain interaction terms between variables. Nevertheless, these can be easily introduced for two dimensions using tensor-product B-splines without an overflowing increase in the number of coefficients. We can then write the additive model with interaction terms as

$$f(x_1, \dots, x_q) = f_{add} + f_{1,2}(x_1, x_2) + \dots + f_{q-1,q}(x_{q-1}, x_q). \tag{1.70}$$

Hence, we use one function $f_i(x_i)$ per input dimension and per interaction term. Using one tensor-product B-spline $t_{j,r}(x_j, x_r)$ for each interaction term, we obtain the model

$$\mathbf{y} = \mathbf{X}_{s_1}\boldsymbol{\beta}_{s_1} + \dots + \mathbf{X}_{s_q}\boldsymbol{\beta}_{s_q} + \sum_{j=1}^{q-1}\sum_{r>j}^{q} \mathbf{X}_{t_{j,r}}\boldsymbol{\beta}_{t_{j,r}} + \boldsymbol{\epsilon}, \tag{1.71}$$

using the tensor-product B-spline basis matrices $\mathbf{X}_{t_{j,r}} \in \mathbb{R}^{n \times d_j d_r}$ and the parameter $\boldsymbol{\beta}_{t_{j,r}} \in \mathbb{R}^{d_j d_r \times 1}$. Using the notation in (1.71), the theoretical framework of linear models can be applied to the additive regression model, since (1.71) can be formulated as linear model yielding

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \begin{bmatrix}\mathbf{X}_{s_1} : & \dots & : \mathbf{X}_{s_q} : \mathbf{X}_{t_{1,2}} : & \dots & : \mathbf{X}_{t_{q-1,q}}\end{bmatrix} \begin{bmatrix}\boldsymbol{\beta}_{s_1} \\ \vdots \\ \boldsymbol{\beta}_{s_q} \\ \boldsymbol{\beta}_{t_{1,2}} \\ \vdots \\ \boldsymbol{\beta}_{t_{q-1,q}}\end{bmatrix} + \boldsymbol{\epsilon} \tag{1.72}$$

with $\mathbf{X} \in \mathbb{R}^{n \times d_{total}}$ as design matrix, $\boldsymbol{\beta} \in \mathbb{R}^{d_{total} \times 1}$ as parameter vector and $d_{total} = \sum_{i=1}^{q} d_i + \sum_{j=1}^{q-1} \sum_{r>j}^{q} d_j d_r$ as total number of parameters in the model. Therefore, the parameters can be calculated efficiently using the Least Squares (LS) algorithm, see Section 1.1.1.1. Further, the assumptions given in Section 1.1 on the error term, as well as on the model function are used [2].

1.3.2 P-Splines

P-splines combine the concepts of B-spline basis functions and regularization to produce smooth function estimations. A function is said to be smooth if its second-order derivative is continuous and does not vary much. Therefore, a penalty of the form

$$\lambda \int (f''(x))^2 \, dx \tag{1.73}$$

is typically introduced to penalize the curvature of a function which is measured by its second-order derivative [12]. The penalty is weighted by the so-called *smoothing parameter* λ. This yields the penalized least squares objective function, cf. Section (1.2.3), as

$$\mathrm{PLS}(\mathbf{y}, \boldsymbol{\beta}; \lambda) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \int (f''(x))^2 \, dx \tag{1.74}$$

using the B-spline basis matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ for some data $D = \{(x^{(i)}, y(i)),\ i = 1, 2, \dots, n\}$. Inserting the B-spline basis function formulation (1.56), using order l and d basis functions, into (1.73) results in

$$\begin{aligned} \int (f''(x))^2 \, dx &= \int \Big(\sum_{j=1}^{d} B_j''(x) \beta_j \Big)^2 \, dx \\ &= \int \sum_{j=1}^{d} \sum_{r=1}^{d} \beta_j \beta_r B_j''(x) B_r''(x) \, dx \\ &= \boldsymbol{\beta}^{\mathrm{T}} \mathbf{K} \boldsymbol{\beta} \end{aligned} \tag{1.75}$$

with the penalty matrix $\mathbf{K}[j, r] = \int B_j''(x) B_r''(x) \, dx$ as a matrix of dimension $\mathbb{R}^{d \times d}$. The entries of $\mathbf{K}$ are given by the integrated products of the second-order derivatives of the B-spline basis functions $B_j(x)$ and $B_r(x)$. For readability, the order l is omitted. These second-order derivatives can be obtained by using the derivative properties of B-spline basis functions given in (1.61) [2].

Eilers and Marx proposed to base the penalty on finite-differences of higher order of the parameters of adjacent B-spline basis functions which circumvents the direct calculation of the derivative and the integral and therefore, reduces the complexity from n, the number of data points to evaluate the integral on, to d, the number of parameters [11]. The squared second-order finite difference gives a good discrete approximation of the integral of the squared second-order derivative in (1.73), i.e.

$$\lambda \sum_{j=3}^{d} (\Delta^2 \beta_j)^2 \approx \lambda \int (f''(x))^2 \, dx, \tag{1.76}$$

where $\Delta^2 \beta_j$ is defined as

$$\begin{aligned} \Delta^2 \beta_j &= \Delta(\Delta \beta_j) \\ &= \Delta(\beta_j - \beta_{j-1}) \\ &= \beta_j - 2\beta_{j-1} + \beta_{j-2}. \end{aligned} \tag{1.77}$$

In matrix form, (1.77) may be given as

$$\mathbf{D}_2 \boldsymbol{\beta} = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_d \end{bmatrix} \tag{1.78}$$

with $\mathbf{D}_2 \in \mathbb{R}^{(d-2) \times d}$. Inserting the matrix form of the second-order finite difference operator (1.78) into (1.76) yields

$$\lambda \sum_{j=3}^{d} (\Delta^2 \beta_j)^2 = \lambda \boldsymbol{\beta}^{\mathrm{T}} \mathbf{D}_2^{\mathrm{T}} \mathbf{D}_2 \boldsymbol{\beta} = \lambda \boldsymbol{\beta}^{\mathrm{T}} \mathbf{K} \boldsymbol{\beta}. \tag{1.79}$$

We then obtain the objective function to minimize as

$$\mathrm{PLS}(\mathbf{y}, \boldsymbol{\beta}; \lambda) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \boldsymbol{\beta}^{\mathrm{T}} \mathbf{K} \boldsymbol{\beta} \tag{1.80}$$

which is equivalent to the objective function of Ridge regression, cf. Section 1.2.3, for the special choice of $\mathbf{K}$ given by $\mathbf{K} = \mathbf{D}_2^{\mathrm{T}} \mathbf{D}_2 \in \mathbb{R}^{d \times d}$. As in Ridge regression, the smoothness parameter λ plays a critical role. For $\lambda \to 0$, the P-spline approaches the underlying B-spline since the penalty term in (1.80) goes to 0. For $\lambda \to \infty$, the P-spline approaches a polynomial model. The order of the polynomial is given by the order of the finite difference penalty, e.g for second-order finite difference penalty, we penalize the discrete approximation of the second-order derivative leading to a linear function, because for these, the second-order derivative is equal to zero. Note that higher-order difference penalties are also possible.

The main advantage of P-splines is their easy set up by replacing the integral of the squared second-order derivative of the B-spline basis functions with the squared second-order finite differences of parameters of adjacent B-spline basis functions. This reduces the computational complexity and allows faster training and evaluation. Hence, P-splines are widely used in practice [11].

A similar penalty term for tensor-product B-splines can be constructed using the Kronecker product. Recall the definition of a tensor-product B-spline given in (1.63) and (1.65) as

$$\begin{aligned} t(x_1, x_2) &= \sum_{j=1}^{d_1} \sum_{r=1}^{d_2} \beta_{j,r} B_{j,r}(x_1, x_2) \\ t(\mathbf{x_1}, \mathbf{x_2}) &= \mathbf{X}\boldsymbol{\gamma}. \end{aligned} \tag{1.81}$$

The spatial alignment of the B-spline basis functions and the corresponding parameters of the two-dimensional tensor-product B-spline needs to be incorporated by the definition of the term *adjacent parameters.* An example for these adjacent parameters, also called spatial neighborhood, is taken from [2] and given in Figure 1.3. Here, we choose the parameters left and right, i.e. $\beta_{j,r-1}$ and $\beta_{j,r+1}$, as well as above and below, i.e. $\beta_{j-1,r}$ and $\beta_{j+1,r}$, as spatial neighborhood for $\beta_{j,r}$.

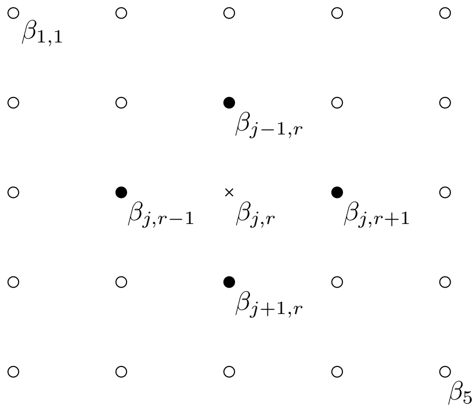

Figure 1.3: Spatial neighborhood or adjacent parameters for a tensor-product B-spline

We now use the concepts of P-splines in both dimensions and penalize the integral of the squared Hessian of the tensor-product B-spline, see (1.73), by a higher-order finite difference approximation in both dimensions. Using second-order finite differences as in (1.76) leads to the following definition of the penalty term

$$\lambda\Big[\sum_{j=3}^{d_1}\sum_{r=1}^{d_2}(\Delta_1^2\beta_{j,r})^2 + \sum_{j=1}^{d_1}\sum_{r=3}^{d_2}(\Delta_2^2\beta_{j,r})^2\Big] \approx \iint (f''(x_1, x_2))^2 \, dx_1 \, dx_2, \tag{1.82}$$

as discrete approximation of the integral of the squared Hessian of the tensor-product B-spline where the first term on the left side calculates the ”row-wise” squared second-order differences using

$$\Delta_1^2\beta_{j,r} = \beta_{j,r} - 2\beta_{j-1,r} + \beta_{j-2,r} \tag{1.83}$$

and the second term on the left side calculates the ”column-wise” squared second-order differences using

$$\Delta_2^2\beta_{j,r} = \beta_{j,r} - 2\beta_{j,r-1} + \beta_{j,r-2}. \tag{1.84}$$

The subscript for Δ in (1.83) and (1.84) indicates the direction of the finite differences. Using the matrix form of the second-order finite difference operator and the Kronecker product, see Appendix A, as well as the parameter vector $\boldsymbol{\gamma} \in \mathbb{R}^{d_1 d_2 \times 1}$, see (1.81), we can then write the "row-wise" penalty as

$$\boldsymbol{\gamma}^{\mathrm{T}}(\mathbf{I}_{d_2} \otimes \mathbf{D}_{1,2})^{\mathrm{T}}(\mathbf{I}_{d_2} \otimes \mathbf{D}_{1,2})\boldsymbol{\gamma} = \sum_{j=3}^{d_1}\sum_{r=1}^{d_2}(\Delta_1^2\beta_{j,r})^2 \tag{1.85}$$

and the "column-wise" penalty as

$$\boldsymbol{\gamma}^{\mathrm{T}}(\mathbf{D}_{2,2} \otimes \mathbf{I}_{d_1})^{\mathrm{T}}(\mathbf{D}_{2,2} \otimes \mathbf{I}_{d_1})\boldsymbol{\gamma} = \sum_{j=1}^{d_1}\sum_{r=3}^{d_2}(\Delta_2^2\beta_{j,r})^2 \tag{1.86}$$

using the identity matrices $\mathbf{I}_{d_1} \in \mathbb{R}^{d_1 \times d_1}$ and $\mathbf{I}_{d_2} \in \mathbb{R}^{d_2 \times d_2}$ and the second-order difference matrices $\mathbf{D}_{1,2} \in \mathbb{R}^{(d_1-2)\times d_1}$ and $\mathbf{D}_{2,2} \in \mathbb{R}^{(d_2-2)\times d_2}$. The first subscript of $\mathbf{D}$ indicates the direction of the finite differences and the second subscript indicates the use of the second-order finite differences. Summing up both penalties leads to a formulation similar to (1.80) given by

$$\mathrm{PLS}(\mathbf{y}, \boldsymbol{\gamma}; \lambda) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\gamma}\|_2^2 + \lambda\boldsymbol{\gamma}^{\mathrm{T}}\mathbf{K}\boldsymbol{\gamma} \tag{1.87}$$

with the tensor-product B-spline basis matrix $\mathbf{X} \in \mathbb{R}^{n \times d_1 d_2}$, the smoothing parameter λ and the penalty matrix $\mathbf{K}$ given by

$$\mathbf{K} = \left[(\mathbf{I}_{d_2} \otimes \mathbf{D}_{1,2})^{\mathrm{T}}(\mathbf{I}_{d_2} \otimes \mathbf{D}_{1,2}) + (\mathbf{D}_{2,2} \otimes \mathbf{I}_{d_1})^{\mathrm{T}}(\mathbf{D}_{2,2} \otimes \mathbf{I}_{d_1})\right] \in \mathbb{R}^{d_1 d_2 \times d_1 d_2}. \tag{1.88}$$

Bibliography

[1] S. N. Wood, *Generalized additive models: an introduction with R.* CRC press, 2017.

[2] L. Fahrmeir, T. Kneib, S. Lang, and B. Marx, *Regression.* Springer, 2007.

[3] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning.* Springer series in statistics New York, 2001, vol. 1, no. 10.

[4] C. M. Bishop, *Pattern recognition and machine learning.* springer, 2006.

[5] D. G. Luenberger, Y. Ye *et al.*, *Linear and nonlinear programming.* Springer, 1984, vol. 2.

[6] V. Blobel and E. Lohrmann, *Statistische und numerische Methoden der Datenanalyse.* Springer-Verlag, 2013.

[7] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[8] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[9] R. L. Eubank and C. H. Spiegelman, "Testing the goodness of fit of a linear model via nonparametric regression techniques," *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 387–392, 1990.

[10] C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor, *A practical guide to splines.* springer-verlag New York, 1978, vol. 27.

[11] P. H. Eilers and B. D. Marx, "Flexible smoothing with b-splines and penalties," *Statistical science*, pp. 89–102, 1996.

[12] F. O'Sullivan, "A statistical perspective on ill-posed inverse problems," *Statistical science*, pp. 502–518, 1986.