Chapter 2 - Fundamentals DRAFT

Weber Jakob

October 16, 2020

Contents

1 Linear Models

1.1 Definition and Model Assumptions

Given the set of data points $\{x_1^{(i)}, \ldots, x_q^{(i)}; y^{(i)}\}$ for $i = 1, \ldots, n$, we aim to model the relation between the input or predictor variables $\{x_1, \ldots, x_p\}$ and the output y with a function $y = f(x_1, \ldots, x_p)$ and an additive noise term ϵ. Thus we obtain the model formulation for the data point i as

$$y^{(i)} = f(x_1^{(i)}, \ldots, x_p^{(i)}) + \epsilon^{(i)} \tag{1}$$

.

The goal is now the estimation of the unknown function f. For this, several assumptions on the model structure are taken:

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

   The function $f(x_1, \ldots, x_p)$ is modeled as a linear combination of inputs, i.e.,

   $$f(x_1, \ldots, x_p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p, \tag{2}$$

   with unknown parameters $\beta_0, \ldots, \beta_p$, which need to be estimated. The model is therefore linear in its parameters as well as in its inputs. **bishop2006patternRecognition** The parameter β_0 is called intercept or bias in the machine learning community. For centered data, i.e. the expected value $\mathbb{E}[x^{(i)}] = 0$, the intercept is equal to zero and can be neglected.

2. *Additive errors*

   The assumptions of additive errors adds the error term ϵ to the unknown function in (2), which leads to the following model structure for the data point i

   $$y^{(i)} = f(x_1^{(i)}, \ldots, x_p^{(i)}) + \epsilon^{(i)}. \tag{3}$$

   This is reasonable for many practical applications, even though it appears quite restrictive.

Commonly, the linear model in (1) is represented in vector notation given by

$$y^{(i)} = f(x_1^{(i)}, \ldots, x_p^{(i)}) = \boldsymbol{x}'\boldsymbol{\beta} + \epsilon^{(i)} \tag{4}$$

where $\boldsymbol{x}' = (1, x_1^{(i)}, \ldots, x_p^{(i)})$ and $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)'$.

To estimate the unknown parameters $\boldsymbol{\beta}$, we define the vectors $\boldsymbol{y} = (y^{(i)}, \ldots, y^{(n)})'$ and $\boldsymbol{\epsilon} = (\epsilon^{(i)}, \ldots, \epsilon^{(n)})'$ as well as the design matrix $\boldsymbol{X}$,

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1q} \\ \vdots & & & \vdots \\ 1 & x_{n1} & \dots & x_{nq} \end{pmatrix} \in \mathbb{R}^{n \times p+1} \tag{5}$$

and generate n equations like Eq.3, which can be combined as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{6}$$

We assume that the design matrix $\boldsymbol{X}$ has full column rank, i.e. $rk(\boldsymbol{X}) = p+1 = q$, implying linear independence of the columns of $\boldsymbol{X}$, which is necessary to obtain a unique estimator for the regression coefficients $\boldsymbol{\beta}$. **fahrmeir2013regression**

Another necessary requirement is that the number of data points n is larger or equal to the number of regression coefficients q, which is equivalent to the statement that the linear system in 6 is not under-determined.

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

1. *Expectation of the error*
   The errors have a mean of zero, i.e. $\mathbb{E}[\epsilon_i] = 0$

2. *Variances and correlation structure of the errors*
   We assume constant error variance with $\mathbb{V}ar[\epsilon_i] = \sigma^2$. This property is called homoscedasticity. Additionally, we assume that the errors are uncorrelated, which means $\mathbb{C}ov[\epsilon_i, \epsilon_j] = 0$ for $i \neq j$. These assumptions combined lead to the covariance matrix $\mathbb{C}ov[\epsilon] = \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}'] = \sigma^2\boldsymbol{I}$.

3. *Gaussian errors*
   The errors follow at least approximately a normal distribution. With assumptions 1 and 2, we obtain that $\epsilon_i = \mathcal{N}(0, \sigma^2)$

It follows from the assumptions on the model function and on the error term that

$$\mathbb{E}[y] = \mathbb{E}[\boldsymbol{x}'\boldsymbol{\beta} + \boldsymbol{\epsilon}] = \boldsymbol{X}\boldsymbol{\beta} \tag{7}$$

$$\mathbb{V}[\boldsymbol{y}] = \mathbb{V}[\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}] = \mathbb{E}\big[(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} - \mathbb{E}[\boldsymbol{y}])^2\big] = \mathbb{V}[\boldsymbol{\epsilon}] = \sigma^2\boldsymbol{I} \tag{8}$$

$$\mathbb{C}ov[y_i, y_j] = \mathbb{C}ov[\epsilon_i, \epsilon_j] = 0, \tag{9}$$

for the mean $\mathbb{E}$ and variance $\mathbb{V}$ of $\boldsymbol{y}$, and the covariance $\mathbb{C}\rtimes\gtrapprox$ between y_i and y_j. With the additionally assumed Gaussian errors, we have

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}). \tag{10}$$

A linear model with multiple input variables can therefore be interpreted as a normal distribution with its mean vector given by $\boldsymbol{\mu} = \boldsymbol{X}\boldsymbol{\beta}$ and its covariance matrix given by $sigma^2\boldsymbol{I}$. To specify the linear model given in (10), we need to estimate the regression coefficients $\boldsymbol{\beta}$ and the variance σ^2.

1.2 Parameter Estimation

The linear model given in Eq. 10 has the unknown parameters $\boldsymbol{\beta}$ and σ which need to be estimated using given data. In the following part, the estimators $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}$ are introduced, and their statistical properties are derived.

1.2.1 Estimation of the Regression Coefficients $\boldsymbol{\beta}$

The two main methods for the estimation of the regression coefficients in the context of linear models are

- Method of Least Squares
- Method of Maximum Likelihood

For Gaussian errors, the maximum likelihood estimator for the regression coefficients coincides with the least squares estimator.

1.2.1.1 The Method of Least Squares

The unknown regression coefficients $\boldsymbol{\beta}$ are estimated by minimizing the sum of squared error

$$\mathrm{LS}(\boldsymbol{y}, \boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 = \sum_{i=1}^{n} \epsilon_i^2 = \boldsymbol{\epsilon}'\boldsymbol{\epsilon} \tag{11}$$

with respect to $\boldsymbol{\beta} \in \mathbb{R}^p$. **friedman2001elements** Rewriting of (11) leads to the least squares criterion

$$\begin{aligned} \mathrm{LS}(\boldsymbol{y}, \boldsymbol{\beta}) &= \boldsymbol{\epsilon}'\boldsymbol{\epsilon} && (12)\\ &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) && (13)\\ &= \boldsymbol{y}'\boldsymbol{y} - 2\boldsymbol{y}'\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta}. && (14) \end{aligned}$$

The least squares criterion is minimized by setting its first derivative equal to zero and by showing that the matrix of second derivatives is positive definite. Applying the rules of differentiation, we obtain

$$\frac{\partial LS(\boldsymbol{y}, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}'\boldsymbol{y} + 2\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta}. \tag{15}$$

The second derivative is given by

$$\frac{\partial^2 LS(\boldsymbol{y}, \boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} = 2\boldsymbol{X}'\boldsymbol{X} \tag{16}$$

Since $\boldsymbol{X} \in \mathbb{R}^{n \times q}$ has, per assumption in Chapter **??**, full rank, the matrix $\boldsymbol{X}'\boldsymbol{X}$ is positive definite. The least squares estimate $\hat{\boldsymbol{\beta}}_{LS}$ is then obtained by solving the so-called *normal equations*

$$\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}'\boldsymbol{y}. \tag{17}$$

Since $\boldsymbol{X}'\boldsymbol{X}$ is positive definite and invertible, the *normal equations* in (17) have a unique solution given by the least squares estimate

$$\hat{\boldsymbol{\beta}}_{LS} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}. \tag{18}$$

1.2.1.2 Maximum Likelihood Estimation

Assuming normally distributed errors, i.e. $\epsilon \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$, the maximum likelihood estimators for the unknown parameters $\boldsymbol{\beta}$ and σ^2 can be computed. **wood2017generalized** Under the normality assumption the likelihood is defined as

$$\mathcal{L}(\boldsymbol{\beta}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\Big(-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\Big). \tag{19}$$

The log-likelihood is then given by

$$l(\boldsymbol{\beta}, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}). \tag{20}$$

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 (11). The maximum likelihood estimator $\hat{\boldsymbol{\beta}}_{ML}$ is therefore equivalent to the least squares estimator $\hat{\boldsymbol{\beta}}_{LS}$ in (18).

1.2.2 Estimation of the Variance σ^2

The estimation of the variance σ^2 is necessary for the construction of confidence intervals of the regression coefficients and for the construction of prediction intervals. It is further used in all kinds of statistical tests. **blobel2013statistische**

1.2.2.1 Maximum Likelihood Estimation

The variance σ^2 can be estimated using the maximum likelihood method by differentiation of the log-likelihood $l(\beta, \sigma^2)$ in 20 with respect to σ^2. The derivative is given by

$$\frac{\partial l(\boldsymbol{\beta}, \sigma^2)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}). \tag{21}$$

Substituting the maximum likelihood estimator $\hat{\boldsymbol{\beta}}_{LS}$ for $\boldsymbol{\beta}$ results in the maximum likelihood estimator σ^2_{ML} for the variance σ^2 given by

$$\hat{\sigma}^2_{ML} = \frac{(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}_{LS})'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}_{LS})}{n} = \frac{\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}}}{n}. \tag{22}$$

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

1.2.2.2 Restricted Maximum Likelihood Estimation

Using $\mathbb{E}[\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}}] = (n-q)\sigma^2$, we obtain for the restricted maximum likelihood estimation of the variance σ^2 the following:

$$\hat{\sigma}^2_{REML} = \frac{1}{n-q}\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}}, \tag{23}$$

which is the commonly used estimator for the variance σ^2. The restricted maximum likelihood estimator for the variance is in general less biased. Therefore, it is a better choice to use this estimator.

1.3 The Hat Matrix

Using the least squares estimator $\hat{\boldsymbol{\beta}}_{LS} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$, we can estimate the mean of $\boldsymbol{y}$ by

$$\widehat{\mathbb{E}[\boldsymbol{y}]} = \hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}_{LS} \tag{24}$$

This results in

$$\hat{\boldsymbol{y}} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{H}\boldsymbol{y}, \tag{25}$$

with the matrix $\boldsymbol{X} \in \mathbb{R}^{n\times n}$, 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

$$\hat{\boldsymbol{\epsilon}} = \boldsymbol{y} - \boldsymbol{y} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}. \tag{26}$$

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

- $\boldsymbol{H}$ is symmetric.
- $\boldsymbol{H}$ is idempotent, i.e. $\boldsymbol{H}^2 = \boldsymbol{H}$.
- The rank of $\boldsymbol{H}$ 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 $i \neq j$.
- The matrix $(\boldsymbol{I} - \boldsymbol{H})$ is also idempotent and symmetric, with $rk(\boldsymbol{I} - \boldsymbol{H}) = n - q$.

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

1.4 Model Selection

Linear models are a widely used technique for regression problems on large data sets $n \gg 0$, since the solution of the *normal equations* in (17) can be computed efficiently for large n. If these data sets also contain a large number of input variables $p \gg 0$, the situation becomes more complicated because possible

interaction effects or correlation of input variables may occur. This 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 some given data set. Model choice criteria, see Chapter 1.4.1 are used to compare different models while model selection techniques, see Chapter 1.4.2 give an algorithmic approach to model creation. Further, we can influence the estimated coefficients $\boldsymbol{\beta}$ directly via regularization, see Chapter 1.4.3.

1.4.1 Model Choice Criteria

One way of comparing various models, i.e models using different sets of variables, is the use of model choice criteria, e.g. Mallow's CP or AIC. Generally, model choice 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 choice criteria are based on the sum of squared prediction error $SPSE$. Therefore, the derivation of the sum of squared prediction error $SPSE$ is given first.

1.4.1.1 Sum of expected Squared Prediction Error

We assume given data $\{x_1^{(i)}, \ldots, x_i^{(p)}; y_i\}$ for $i = 1, \ldots, n$. Further, we assume that the expectation $\mathbb{E}[y_i] = \mu_i$ and the variance $\mathbb{V}ar[y_i] = \sigma^2$. Using the q variables x_j we can generate the corresponding design matrix $\boldsymbol{X}$ for the linear model as in (6), i.e.

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} \tag{27}$$

with $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{X} in \mathbb{R}^{n \times q}$ and $\boldsymbol{\beta} \in \mathbb{R}^q$. The least squares estimator for $\boldsymbol{\beta}$ is then given by

$$\hat{\boldsymbol{\beta}}_{LS} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}. \tag{28}$$

The data $\boldsymbol{y}$ can be interpreted as random variable. We can then define an estimator $\hat{\boldsymbol{y}}$ for the vector $\boldsymbol{\mu}$ of expectations $\mu_i = \mathbb{E}[y_i]$ by

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}. \tag{29}$$

The following properties for $\hat{\boldsymbol{y}}$ hold:

- $\mathbb{E}[\hat{\boldsymbol{y}}] = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\mathbb{E}[\boldsymbol{y}]$
- $\mathbb{C}ov[\hat{\boldsymbol{y}}] = \sigma^2\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$
- $\sum_{i=1}^n \mathrm{Var}[\hat{y}^{(i)} i] = \sigma^2 \mathrm{tr}(\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}') = |M|\sigma^2$, where M represents the number of used variables.

- Sum of Mean Squared Error

$$\begin{aligned}\text{SMSE} &= \sum_{i=1}^{n} \mathbb{E}[\hat{y}^{(i)} - \mu^{(i)}]^2 \\ &= \sum_{i=1}^{n} \mathbb{E}\big[(\hat{y}^{(i)} - \mathbb{E}[\hat{y}^{(i)}]) + (\mathbb{E}[\hat{y}^{(i)}] - \mu^{(i)})\big]^2 \\ &= |M|\sigma^2 + \sum_{i=1}^{n} (\mathbb{E}[\hat{y}^{(i)}] - \mu^{(i)})^2.\end{aligned} \tag{30}$$

If we now assume new data

$$\{x_1^{(i)}, \ldots, x_p^{(i)};\ y^{(n+i)} = \mu^{(i)} + \epsilon^{(n+1)}\} \quad \text{for } i = 1, \ldots, n, \tag{31}$$

we can use the estimator $\hat{\boldsymbol{y}}$ as a prediction for these new observations. We can then derive the sum of the expected squared prediction errors SPSE, given by

$$\begin{aligned}\text{SPSE} &= \sum_{i=1}^{n} \mathbb{E}[y^{(n+i)} - \hat{y}^{(i)}]^2 \\ &= \sum_{i=1}^{n} \mathbb{E}\big[(y^{(n+i)} - \mu^{(i)}) - (\hat{y}^{(i)} - \mu^{(i)})\big]^2 \\ &= \sum_{i=1}^{n} \mathbb{E}[y^{(n+i)} - \mu^{(i)}]^2 + 2\mathbb{E}[(y^{(n+i)} - \mu^{(i)})(\hat{y}^{(i)} - \mu^{(i)})] + \mathbb{E}[\hat{y}^{(i)}] - \mu^{(i)}]^2 \\ &= \sum_{i=1}^{n} \mathbb{E}[y^{(n+i)} - \mu^{(i)}] + \sum_{i=1}^{n} (\mathbb{E}[\hat{y}^{(i)}] - \mu^{(i)})^2 \\ &= n\sigma^2 + \text{SMSE} \\ &= n\sigma^2 + |M|\sigma^2 + \sum_{i=1}^{n} (\mathbb{E}[\hat{y}^{(i)}] - \mu^{(i)})^2.\end{aligned} \tag{32}$$

The sum of the expected squared prediction error can be split into three parts

- *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.
- *Variance 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 using a smaller number of variables.
- *Squared Bias Term*: $\sum_{i=1}^{n} (\mathbb{E}[\hat{y}^{(i)} - \mu^{(i)}])^2$
  The last term can be interpreted as bias. It can be reduced by increasing the model complexity.

The sum of expected squared prediction error is an example of the bias-variance trade-off, which is characteristic for all statistical models. It states that by increasing model complexity, the bias is reduced but instead the variance is increased. On the other side, by decreasing model complexity, the variance of the model is reduced, but the bias is increased. **bishop2006patternRecognition**

In practice, the true value for the SPSE is not accessible since μ_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

$$\widehat{\text{SPSE}} = \sum_{i=1}^{n} (y^{(n+i)} - \hat{y}^{(i)})^2. \tag{33}$$

   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 error term depending on the estimated variance and the model complexity. The estimate is thus given by

$$\widehat{\text{SPSE}} = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2 + |M|\hat{\sigma}^2. \tag{34}$$

Typically used model choice criteria follow the basic idea of the SPSE. **fahrmeir2013regression**

1.4.1.2 Corrected Coefficient of Determination

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

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \hat{\epsilon}^{(i)} \hat{\epsilon}^{(i)}}{\sum_{i=1}^{n} (y^{(i)} - \bar{y})^2}, \tag{35}$$

where $\bar{y}$ is defined as the mean value of $\boldsymbol{y}$. The major drawback of R^2 is that it will never decrease when further input variables 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

$$R^2_{corr} = 1 - \frac{n-1}{n-q}(1 - R^2). \tag{36}$$

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. **fahrmeir2013regression**

1.4.1.3 Mallow's Cp

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

$$\mathrm{C}_p = \frac{\sum_{i=1}^{n}(y^{(i)} - \mathbb{E}[y^2(i)])^2}{\hat{\sigma}^2} - n + 2|M|, \tag{37}$$

where M is again the number of used parameters. Lower values of Mallow's C_p correspond to a better model fit. **fahrmeir2013regression**

1.4.1.4 Akaike Information Criterion

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

$$\mathrm{AIC} = -2l(\hat{\boldsymbol{\beta}}_{ML}, \hat{\sigma}^2_{ML}) + 2(|M| + 1) \tag{38}$$

where $l(\hat{\boldsymbol{\beta}}_{ML}, \hat{\sigma}^2_{ML})$ is the value of the log-likelihood at its maximum and $|M|$ is the number of used parameters. We again have the standard model choice criteria structure of a data dependent term, here the maximal log-likelihood, and a model dependent term given.

The log-likelihood for a linear model assuming Gaussian errors is given by

$$-2l(\hat{\boldsymbol{\beta}}_{ML}, \hat{\sigma}^2_{ML}) = n\log(\hat{\sigma}^2_{ML}) + n. \tag{39}$$

Therefore, neglecting the constant value n, the AIC evaluates to

$$\mathrm{AIC} = n\log(\hat{\sigma}^2_{ML}) + 2(|M| + 1). \tag{40}$$

Lower values of AIC correspond to a better model fit. **fahrmeir2013regression**

1.4.1.5 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

$$\mathrm{BIC} = -2l(\hat{\boldsymbol{\beta}}_{ML}, \hat{\sigma}^2_{ML}) + \log(n)(|M| + 1). \tag{41}$$

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

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

Lower values of BIC correspond to a better model fit. **fahrmeir2013regression**

1.4.1.6 Cross Validation

The basic idea of cross validations is to split the existing data set into multiple smaller ones and to fit one model to each of these smaller data sets. These models are then evaluated by the calculation of the SPSE on the data which it was not trained on. The model which has the smallest error is then chosen as final estimate.

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 $\boldsymbol{y}$ and the *hat matrix* $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$. The cross validation score is then given by

$$\mathrm{CV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y^{(i)} - \hat{y}^{(i)}}{1 - h_{ii}}\right)^2. \tag{42}$$

where h_{ii} denotes the diagonal elements of the *hat matrix* and $\hat{y}^{(i)}$ is defined as the prediction of the model for the input $\{x_1^{(i)}, \ldots, x_p^{(i)}\}$. A lower cross validation score corresponds to a better model fit. **golub1979**

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 H is numerically expensive to compute. In the GCV score, the diagonal elements of the hat matrix h_{ii} are replaced by the mean of the trace of H. The GCV score is then given by

$$\mathrm{GCV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y^{(i)} - \hat{y}^{(i)}}{1 - \mathrm{trace(H)}/n}\right)^2. \tag{43}$$

The numerical advantage comes from the fact that the trace of a product of matrices is not changed when cyclically permuting the product of matrices, i.e.

$$\mathrm{trace}(\boldsymbol{H}) = \mathrm{trace}(\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}') = \mathrm{trace}(\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}). \tag{44}$$

The trace can therefore be computed from the product of two matrices of shape $q \times q$. **fahrmeir2013regression**

1.4.2 Model Selection Techniques

To make use of the various model choice criteria, some algorithmic approach to model selection needs to be given. The most commonly used are given below. We always start with a candidate model. **fahrmeir2013regression**

1.4.2.1 Forward Selection

We start with a candidate model which includes a small number of variables. In each iteration of forward selection, an additional variable is included into

the candidate model. The added variable is the one with leads to the largest reduction of a predefined model choice criteria. The algorithm stops, if no further reduction is achieved.

1.4.2.2 Backward selection

We 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 choice criteria. The algorithm stops, if no further reduction is possible.

1.4.2.3 Step-wise Selection

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.

1.4.3 Regularization

Model selection can also be achieved using regularization techniques by directly influencing the coefficients $\boldsymbol{\beta}$ which need to be estimated given some data. 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 in (11). This leads to the penalized least squares criterion

$$\text{PLS}(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda \text{pen}(\boldsymbol{\beta}) \tag{45}$$

where λ is the so-called smoothing parameter and $\text{pen}(\boldsymbol{\beta})$ is the penalty term describing the regularization technique. The two most commonly used forms of regularization are Ridge regression and Lasso regression. Both are explained in detail in the following.

1.4.3.1 Ridge Regression

In Ridge regression, the penalty term in the penalized least squares criterion in (45) is given by the squared L_2-norm of the coefficient vector $\boldsymbol{\beta}$. The objective function to minimize is therefore given by

$$\text{PLS}_{Ridge}(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda \boldsymbol{\beta}'\boldsymbol{\beta} \tag{46}$$

with the closed form solution

$$\hat{\boldsymbol{\beta}}_{PLS,r} = \arg\min_{\boldsymbol{\beta}} \text{PLS}_{Ridge} = (\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}'\boldsymbol{y}, \tag{47}$$

where $\boldsymbol{I} \in \mathbb{R}^{q \times q}$ is the identity matrix. The additional penalty term in Ridge regression leads to smaller parameter estimates $\hat{\boldsymbol{\beta}}_{PLS,r}$ compared to the un-penalized estimate $\hat{\boldsymbol{\beta}}_{LS}$. For large values of the smoothing parameter λ, the parameter estimates will be converge towards, but never reach, zero.

Ridge regression is commonly when the input dimension p is high. **hoerl1970ridge**

1.4.3.2 Lasso Regression

One drawback of Ridge regression is that it does not produce sparse solutions, i.e. all estimated coefficients will be different from zero. Lasso regression tackles this problem by the use of the L_1-norm as penalty term. The penalized least squares objective function is then given by

$$\mathrm{PLS}_{Lasso}(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\| + \lambda \sum_{i=1}^{k} |\beta_i|, \tag{48}$$

where λ again plays the role of a smoothing parameter. Therefore, we need to solve the optimization problem

$$\hat{\boldsymbol{\beta}}_{PLS,l} = \arg\min_{\boldsymbol{\beta}} \mathrm{PLS}_{Lasso}. \tag{49}$$

No closed form solution for the problem given in (49) is available. The Lasso estimates $\hat{\boldsymbol{\beta}}_{PLS,l}$ are calculated using either quadratic programming techniques, given in **tibshirani1996lasso**, or least angle regression, given in **efron2004leastangleregression**.

Ridge regression penalized large coefficients much stronger than Lasso regression due the the quadratic penalty, while for small coefficient values, the Lasso penalty has much more influence. **tibshirani1996lasso**

2 Splines

A spline is a piece-wise polynomial defined on a sequence of knots. This definition is quite general. Therefore, there exist a large variety of splines, ranging from regression splines in **eubank1990regressionsplines**, over B-splines in **deBoor1978practicalGuideToSplines** to natural cubic splines and many more. We will focus on the definition of B-splines and tensor-product splines, as the multi-dimensional expansion of B-splines, as well as P-splines. **deBoor1978practicalGuideToSplines eilers1996flexible**

2.1 B-Splines

We lay the focus on the definition and use of B-splines, which are constructed from polynomial pieces in a recursive manner. Given the knot sequence $K = \{k_i, \ldots, k_{i+m+2}\}$, the B-spline $B_i^m(x)$ of degree m is defined by means of the Cox-de Boor recursion formula as

$$B_i^0(x) = \begin{cases} 1, & \text{if} \quad k_i \leq x < k_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{50}$$

$$B_i^m(x) = \frac{x - k_i}{k_{i+m} - k_i} B_i^{m-1}(x) + \frac{k_{i+m+1} - k}{k_{i+m+1} - k_{i+1}} B_{i+1}^{m-1}(x). \tag{51}$$

Therefore, $B_j^0(x)$ is the zero function, except for $x \in [k_j, k_{j+1}]$ where it is equal to 1. $B_j^1(x)$ is then the known *hat function*, being zero except for $x \in [k_j, k_{j+2}]$.

Figure 1 shows an example of a B-spline of degree $m = 1$ on the left. It consists of two linear pieces, one defined from k_1 to k_2 and the other from k_2 to k_3. Everywhere else, the B-spline is equal to zero. At the joining points, the values of the polynomial pieces are equal. This locality is a very attractive feature of B-splines. In the right part of Figure 1, a B-spline of degree $m = 2$ is shown, which consist of three quadratic pieces, defined on the knot sequence $\{k_1, k_2, k_3, k_4\}$. At the joining points, the values as well as the first derivatives of the quadratic pieces are equal.

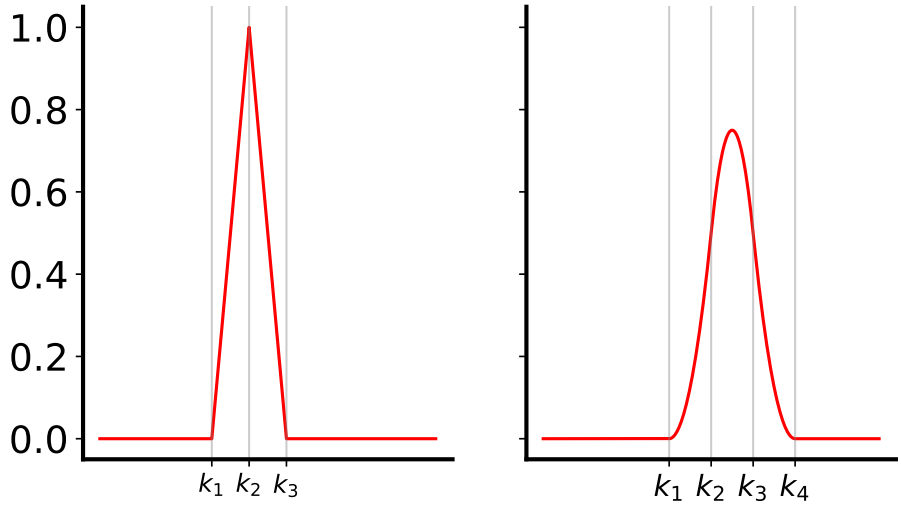

Figure 1: Linear and Quadratic Spline

The general properties of a B-spline of degree m are the following:

- It consists of $m + 1$ polynomial pieces of degree m, e.g. a cubic spline ($m = 3$) consists of 4 cubic pieces.
- The pieces join at m inner knots.
- At these knots, the derivatives up to order $m - 1$ are continuous.
- The B-spline is positive on the domain spanned by $m+2$ knots, everywhere else it is zero, e.g. for $m = 2$, a sequence of 4 knots is necessary.
- At every given x, only $m + 1$ B-splines are non-zero.

The collection of k B-splines of degree m over a sequence of $k + 2(m - 1)$ knots is called B-spline basis. The $2(m-1)$-knots are the boundary knots while the k knots are the interior knots. The knots can either be an equidistant sequence, which facilitates the construction and estimation of the coefficients, or a non-equidistant sequence.

A smooth function $f(x)$ can then be represented using the basis function approach given by

$$f(x) = \sum_{i=1}^{k} B_i^m(x)\beta_i = \boldsymbol{x}'\boldsymbol{\beta} \tag{52}$$

using the B-spline basis functions $B_i^m(x)$ of degree m and the coefficients $\boldsymbol{\beta} \in \mathbb{R}^k$. The basis functions can also be given in vector notation as $\boldsymbol{x}' = (B_1^m(x), \ldots, B_k^m(x))$.

The use of B-splines as basis functions for uni-variate and non-parametric regression is very attractive. A linear combination of cubic B-splines gives a smooth curve, i.e. first and second order derivatives are continuous. A further advantage of B-splines is that once the basis is given, the coefficients can be estimated using the Least Squares algorithm as in (11).

Given a set of data points $\{x^{(i)}, y^{(i)}\}$ for $i = 1, \ldots, n$, the least squares objective function for B-splines is the given by

$$Q_{bsplines}(\boldsymbol{y}, \boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 \tag{53}$$

for the B-spline basis $\boldsymbol{X} \in \mathbb{R}^{n \times k}$ for n data points and k splines. Solving the optimization problem, i.e.

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} Q_{bsplines}(\boldsymbol{y}, \boldsymbol{\beta}) \tag{54}$$

leads to the estimated coefficients

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}. \tag{55}$$

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 calculate e.g confidence intervals for the regression coefficients and the prediction.

B-splines of appropriate order $m > 2$ produce smooths curves, where the smoothness is mostly determined by the number of splines used. For a low number, the curve will be quite smooth, but possess a large data error. When using a high number of splines, 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 machine learning. It is therefore necessary to introduce some kind of regularization. **deBoor1978practicalGuideToSplines**

2.2 P-Splines

P-splines use the concepts of regularization to produce smooth function estimations. They where introduced by Eilers and Marx in **eilers1996flexible** to tackle the problem introduced above. Eilers and Marx simplified and generalized the idea of **osullivan1986statistical**, who introduced a smoothness penalty on the integral of the squared second derivative of the estimated spline to penalized

wiggly function estimates. They proposed to use equidistant knots and penalty based on finite differences of order d of the coefficients of adjacent B-splines. The finite differences of order $d = 1$ for p splines acting on the coefficient vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)'$ are given by

$$\Delta^1 \boldsymbol{\beta} = \sum_{j=2}^{k-1} \beta_j - \beta_{j-1} \tag{56}$$

and in matrix form

$$\Delta^1 = \boldsymbol{D}_1 = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & & \end{pmatrix} \in \mathbb{R}^{k-1 \times k}. \tag{57}$$

This leads to the penalized least squares formulation, similar to (45),

$$Q_{pslines}(\boldsymbol{y}, \boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda_s \mathcal{J}_s(\boldsymbol{\beta}; d) \tag{58}$$

where $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$ is the mean squared error on the data for the spline fit, $\mathcal{J}_s(\boldsymbol{\beta}; d) = \boldsymbol{\beta}^T \boldsymbol{D}_d^T \boldsymbol{D}_d \boldsymbol{\beta}$ is the smoothness penalty term given by the matrix form of the difference operator Δ^d of order d and λ_s is the smoothness parameter determining the effect of the smoothing penalty. The estimated coefficients are then given by the minimization of

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} Q_{psplines}(\boldsymbol{y}, \boldsymbol{\beta}) \tag{59}$$

as

$$\boldsymbol{\beta} = (\boldsymbol{X}^T \boldsymbol{X} + \lambda_s \boldsymbol{D}_d^T \boldsymbol{D}_d)^{-1} \boldsymbol{X}^T \boldsymbol{y}. \tag{60}$$

The main advantage of P-splines is their easy set up. This advantage is diminished if uneven knot placement is chosen.

2.3 Tensor-Product Splines

Tensor-product splines can be seen as the multi-dimensional extension of univariate B-splines. We start with a B-spline basis for each dimension and construct the tensor-product spline from these.

We examine an example for two input dimensions x_1 and x_2. Assume that we have B-spline bases available, i.e. $\boldsymbol{A} \in \mathbb{R}^{n \times k}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times k}$ using the same number of splines k for both bases, for representing the functions $f_1(x_1)$ and $f_2(x_2)$ given by

$$f_1(x_1) = \sum_{i=1}^{k} \alpha_i a_i(x_1) = \boldsymbol{A}\boldsymbol{\alpha}, \tag{61}$$

$$f_2(x_2) = \sum_{j=1}^{k} \beta_j b_j(x_2) = \boldsymbol{B}\boldsymbol{\beta} \tag{62}$$

where $\alpha_i \in \mathbb{R}^k$ and $\boldsymbol{\beta}_j \in \mathbb{R}^k$ are the coefficients and $a_i(x_1)$ and $b_j(x_2)$ are the known basis functions. To allow the function $f_1(x_1)$ to smoothly vary with x_2, its coefficients α_i must vary smoothly with x_2. By using the already available basis for representing smooth functions in x_2, we can write

$$\alpha_i(x_2) = \sum_{j=1}^{k} \beta_{ij} b_j(x_2) \tag{63}$$

which leads to

$$f_{1,2}(x_1, x_2) = \sum_{i=1}^{k} \sum_{j=1}^{k} \beta_{ij} b_j(x_2) a_i(x_1). \tag{64}$$

We can therefore combine the individual basis matrices to generated a new basis matrix for the tensor-product spline. For any set of data $(x_1^{(i)}, x_2^{(i)})$ for $i = 1, \ldots, n$, the relationship between the model matrix $\boldsymbol{X}$ of the tensor-product smooth and the marginal model matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is given by

$$\boldsymbol{X} = \boldsymbol{A} \otimes \boldsymbol{B} \tag{65}$$

where $\otimes$ indicates the use of the Kronecker product, $\boldsymbol{X} \in \mathbb{R}^{n \times k^2}$ denotes the tensor-product spline basis, $\boldsymbol{A} \in \mathbb{R}^{n \times k}$ denotes the B-spline basis for dimension 1 and $\boldsymbol{B} \in \mathbb{R}^{n \times k}$ denotes the B-spline basis for dimension 2. The model matrix for the tensor-product smooth is therefore given by the Kronecker product of the marginal model matrices. **wood2006GAM**

This approach can in theory be continued for as much input dimensions as required. In practice, modeling more than two input dimensions using tensor-product splines becomes infeasible because if the enormous increase of basis functions and therefore coefficients to estimate. A smoothness penalty term for tensor-product splines can also be constructed using the Kronecker product. Further explanations are given in Chapter 3.

3 Structured Additive Regression

We have again some given data $\{x_1^{(i)}, \ldots, x_q^{(i)}; y^i\}$ for $i = 1, \ldots, n$ and want to model the generally non-linear relationship between the input data $\{x_1^{(i)}, \ldots, x_q^{(i)}\}$ and the output $y^{(i)}$ by some multi-dimensional function $f(x_1, \ldots, x_q)$. Using, e.g. high-dimensional tensor-product splines, to model the function is computationally expensive, since the number of regression coefficients increases exponentially.

To circumvent this problem, we now assume the restrictive structure of additive models, given by

$$f(x_1, \ldots, x_q) = f_1(x_1) + \cdots + f_q(x_q). \tag{66}$$

Hence, we use one smooth function $f_i(x_i)$ per input dimension and assume an additive structure. **fahrmeir2013regression** Using the concepts introduced in Chapter 2, we obtain for each smooth function a linear model

$$f_i(x_i) = \boldsymbol{X}_i \boldsymbol{beta}_i \tag{67}$$

where $\boldsymbol{X}_i \in \mathbb{R}^{n \times k_i}$ is the B-spline basis using k_i splines for the smooth function $f_i(x_i)$ of input dimension i and $\boldsymbol{\beta}_i \in \mathbb{R}^{k_i}$ are the coefficients to be estimated. We can also use the already described penalization approaches given in Chapter 2.2.

The model given in (66) does not contain interaction terms between variables. Nevertheless, these can be easily introduced for 2 dimensions using tensor-product splines without an overflowing increase of the number of coefficients.

We can then write the structured additive model in matrix notation as

$$\boldsymbol{y} = \boldsymbol{X}_1 \boldsymbol{\beta}_1 + \cdots + \boldsymbol{X}_q \boldsymbol{\beta}_q + \sum_{i=1}^{n_{interact}} \boldsymbol{X}_{tps,i} \boldsymbol{\beta}_{tps,i} + \boldsymbol{\epsilon} \tag{68}$$

using the error term $\boldsymbol{\epsilon}$ and $n_{interact}$ as the number of interactions to include via tensor-product spline bases $\boldsymbol{X}_{tps,i}$. This can be solved using ordinary least squares. If we choose to include penalization, we can solve this using penalized least squares.

Using the notation in (68) the theoretical framework of linear models can be applied to structured additive regression models. Therefore, the assumptions given in Chapter 1.1 on the error term, as well as on the model functions are used. **fahrmeir2004penalized**