Diplomarbeit

Generalized Additive Models: Background, Definitions, Extensions

Zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Masterstudium Statistik-Wirtschaftsmathematik

eingereicht von

Christopher Rieser, BSc MSc

Matrikelnummer 0815803

ausgeführt am Institut für Stochastik und Wirtschaftsmathematik
der Fakultät für Mathematik und Geoinformation der Technischen Universität Wien

Betreuer: Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser

Wien, 31. August 2018:

Unterschrift Verfasser

Unterschrift Betreuer

Acknowledgements

I would like to express my sincere thanks to my supervisor Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser for all his help in guiding me in the process of finding the right topic and for all his comments, remarks and valuable time spent on discussions and proofreading.
Furthermore, a very special thanks also goes to my mother Beatrice who has supported me all these years and who has been of so much help. I've always considered her a role model.
Also, much thanks goes to my girlfriend Marlene. Her understanding, encouragement and love was endless these last years.
At last, I also want to thank my father Daly and my two brothers Patrick and Daniel. They have supported me a lot these last years.

Preface

This thesis gives an extensive overview of generalized additive models with an outlook on their robustification. Chapter 1 shortly introduces the main problems. Chapter 2 considers additive models and generalized linear models. In Chapter 3 additive models and generalized linear models will be merged to what is known as generalized additive models. Furthermore, we will discuss one way of robustifying the latter and end with three illustrations of GAMs.

Contents

Notation

$\mathbb{N}$	$\{1, 2, 3, ...\}$
$\mathbb{N}_0$	$\{0, 1, 2, ...\}$
$\mathcal{H}$	A Hilbert space
$\mathbf{x}$	A vector
$\mathbf{X}$	A matrix
$\mathbf{X}'$	The transposed of matrix $\mathbf{X}$
x_i	The i-th element of $\mathbf{x}$
$\mathbf{X}_{ij}$	ij-th Element of matrix $\mathbf{X}$
$\mathbf{X}_{:,i}$	The i-th column of $\mathbf{X}$
$\mathbf{X}_{i,:}$	The i-th row of $\mathbf{X}$
$\mathbf{X}_{m:n,r:s}$	Matrix consisting of elements x_{ij}, with $m \leq i \leq n$ and $r \leq j \leq s$
$(x_i)_:$	A vector with elements x_i
$(x_{ij})_{:,:}$	A matrix defined by the elements x_{ij}
$\mathbf{D}$	A diagonal matrix
$\sum_k a_k h_k(\cdot)$	A linear combination of the functions $h_k(\cdot)$
$\sum_{k=0}^{d} a_k x^k$	Polynomial of degree d
$\lvert\boldsymbol{\alpha}\rvert$	Abbreviation for $\alpha_1 + ... + \alpha_p$
$\boldsymbol{a}_{\boldsymbol{\alpha}}$	Abbreviation for $a_{\alpha_1,...,\alpha_p}$
$\boldsymbol{x}^{\boldsymbol{\alpha}}$	Abbreviation for $x^{\alpha_1} \cdots x^{\alpha_p}$
$\sum_{\lvert\boldsymbol{\alpha}\rvert \leq d} \boldsymbol{a}_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$	Polynomial in p variables of degree d
$\mathbb{P}_d[\boldsymbol{x}]$	The space of all polynomials of degree d
∇f	The gradient of the function f
$\mathcal{H}_f$	The Hessian of the function f
$L^2(\Omega)$	The space $\{f : \Omega \to \mathbb{R} \,\lvert f \text{ is measurable}, \int f^2 < \infty\}$
Y	A random variable
y	A realization of the random variable Y
$\mathbb{E}(X)$	The expectation of the random variable X
$\mathbb{V}\text{ar}(X)$	The variance of the random variable X
$Y\lvert X$	Y conditional on X
$\mathbb{E}(Y\lvert X)$	The conditional expectation of X conditional on Y
$\mathbf{1}_A(\cdot)$	The indicator function for a measurable set A

CHAPTER 1

Some motivation

In this chapter we will quickly motivate additive models and generalized linear models, denoted from now on as AM and GLM, as a generalization of a simple linear model with Gaussian error. All considerations in this chapter are heuristical and lack mathematical exactness. This is done on purpose to highlight motivations.

Assume that we have observations

$$(y_1, x_{11}, ..., x_{1p}), ..., (y_n, x_{n1}, ..., x_{np})$$

and we suspect that these fulfill the linear relation

$$y_i = a_0 + a_1 x_{i1} + ... + a_p x_{ip} + \epsilon_i, \tag{1.1}$$

for $i \in \{1, ..., n\}$, where $a_0, ..., a_p$ are unknown parameters and ϵ_i are independent Gaussian distributed variables with mean zero and unknown constant variance - representing the error in a measurement. The goal would be then to estimate the parameters.

Astonishingly, this simple model works quite well in some situations, however there are two major drawbacks which we can directly see:

Firstly, as all the explanatory variables contribute in a linear way, no interactions between variables or higher order terms, e.g. $x_1^2 x_4^3$, are considered. This can result in a poor model, because many times interactions and higher order terms are necessary to get good results. One natural extension would therefore be to consider interactions and higher order terms, but this is computationally very expensive and might not be feasible. Looking at the case of a polynomial in p variables having degree d, i.e. $\sum_{|\boldsymbol{\alpha}| \leq d} \boldsymbol{a}_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$, where we have

used the multiindex notation, we can deduce that the number of coefficients is equal to $\binom{p+d}{d}$. This number grows very quickly in d and p and for high dimensions, higher order interactions are thus impossible to consider. Such a situation is known as the curse of dimensionality.

Secondly, although the assumption of the errors being normally and independently distributed, with constant variance - thus Y being normally and independently distributed with constant variance - holds in many cases, there are also many examples where this is simply not true and so this is a major limitation of our model. Y could, for example, be Poisson or binomially distributed and, as we will see in Chapter 2, the model assumption (1.1) would not make much sense.

In the following second chapter we will see a remedy for each of these two problems by considering AMs and GLMs. After this we will unify, in the third chapter, the latter two to what is known as generalized additive models (GAM).

CHAPTER 2

A recap of additive models and generalized linear models

Before we start to talk about AMs let us put the situation of Chapter 1 on mathematical solid grounds.

Assume that we have a real valued random vector $(Y|(X_1, ..., X_p), X_1, ..., X_p)$ on some probability space $(\Omega, \mathcal{A}, \mathbb{P})$. We would then like to be able to predict, for each realization ω, $Y|X(\omega) \in \mathbb{R}$ by some function of $X(\omega)$ - where we used the abbreviation $X = (X_1, ..., X_p)$. This means that we want to find a measurable function $f : \Omega \to \mathbb{R}$, such that $f(X(\omega))$ gets "close" to $Y|X(\omega)$. One way to measure "closeness" could be to take the Euclidean norm - on the one hand as it is the natural norm to work with in $\mathbb{R}$, and on the other as it has good smoothness properties. Therefore we would like to find a function f which minimizes the quantity $(Y|X(\omega) - f(X(\omega)))^2$. As we are in the setting of a probability space, we thus end up with the following problem; where we omitted writing the conditional dependence of Y on X, hence only writing Y instead of $Y|X$:

$$\min_{f \in \mathcal{H}} \mathbb{E}\left(\left(Y - f(X)\right)^2\right). \tag{2.1}$$

This problem only makes sense if we also assume that Y is in $L^2(\Omega)$ and that the space $\mathcal{H}$ is the space of all measurable functions f for which the quantity $f(X)$ is in $L^2(\Omega)$.

It turns out, and it is not hard to show, that the function minimizing (2.1) is given by $\mathbb{E}(Y|X)$. Doing the following

$$
\begin{aligned}
&\mathbb{E}\left(\left(Y - f(X)\right)^2\right) \\
&= \mathbb{E}\left(\mathbb{E}\left(\left(Y - f(X)\right)^2 \middle| X\right)\right) \\
&= \mathbb{E}\left(\mathbb{E}\left(\left(Y - \mathbb{E}(Y|X) + \mathbb{E}(Y|X) - f(X)\right)^2 \middle| X\right)\right) \\
&= \mathbb{E}\left(\mathbb{E}\left(\left(Y - \mathbb{E}(Y|X)\right)^2 \middle| X\right)\right) + \mathbb{E}\left(\mathbb{E}\left(\left(\mathbb{E}(Y|X) - f(X)\right)^2 \middle| X\right)\right) \\
&\quad + 2\,\mathbb{E}\left(\mathbb{E}\left(\left(\left(Y - \mathbb{E}(Y|X)\right)\left(\mathbb{E}(Y|X) - f(X)\right)\right) \middle| X\right)\right) \\
&= \mathbb{E}\left(\left(Y - \mathbb{E}(Y|X)\right)^2\right) + \mathbb{E}\left(\left(\mathbb{E}(Y|X) - f(X)\right)^2\right) \\
&\quad + 2\,\mathbb{E}\left(\mathbb{E}\left(\left(Y - \mathbb{E}(Y|X)\right) \middle| X\right)\left(\mathbb{E}(Y|X) - f(X)\right)\right) \\
&= \mathbb{E}\left(\left(Y - \mathbb{E}(Y|X)\right)^2\right) + \mathbb{E}\left(\left(\mathbb{E}(Y|X) - f(X)\right)^2\right),
\end{aligned}
$$

where we used $\mathbb{E}\left(\left(Y - \mathbb{E}(Y|X)\right)\middle|X\right) = 0$, we can see that the first term is constant and doesn't include f and the second term is always positive. This means that the minimizing function is $f(X) = \mathbb{E}(Y|X)$.

Knowing that problem (2.1) is solved by $f(X) = \mathbb{E}(Y|X)$ is however not really useful, as to estimate $\mathbb{E}(Y|X)$ appropriately we would need for each x multiple measurements - which are usually not given - even if we knew the family of distributions for $Y|X$ that is underlying.
This forces us to assume that $\mathbb{E}(Y|X)$ has a functional form or at least can be approximated reasonably by one - meaning that $\mathbb{E}(Y|X) \approx f_\theta(X)$ for some function f_θ depending on some parameter $\theta \in \Theta$, e.g. a powerseries in X - and go from there, to avoid needing multiple realizations of y for each x.

In this spirit let us go back to the original problem (2.1) and now only look for $f_\theta \in \mathcal{H}$, with $\theta \in \Theta$, which minimize the objective of (2.1). By the same arguments as before we could show that a function f_θ minimizing (2.1) actually minimizes the L^2 distance of f_θ and $\mathbb{E}(Y|X)$; thus finding such an f_θ is equivalent to finding an f_θ that approximates $\mathbb{E}(Y|X)$.

We still need to make some choice now about the functional form of f_θ. A good start could be to consider the space of polynomials of a certain fixed degree; because hopefully as all continuous functions on a bounded set are arbitrarily well approximable by polynomials this will give us good results.
We will start a bit more general and consider $\mathcal{H}$ to be a finite dimensional linear space - linear because it makes things much easier - i.e every f has the form $\sum_{k=1}^{m} a_k h_k(\cdot)$, where the a_k are the parameters to be estimated and $h_k(\cdot)$ are m functions that span the whole space $\mathcal{H}$. All in all, this means

that we try to solve the following problem

$$\min_{a_k} \mathbb{E}\left(\left(Y - \sum_k a_k h_k(X)\right)^2\right).$$

Theoretically we could write down the solution to this problem directly by differentiating in a_k and setting the derivatives to zero - as this is a convex function in a_k this always gives us the unique minimum. However the solution would require us to calculate moments of $h_k(X)$. Usually we have no idea about the distribution of X, which can be very high dimensional and thus it also makes no sense to look at the sample distribution. So this is not possible.

A solution to this could be to plug in the sample distribution constructed by our datapoints

$$(Y_1|X_1), ..., (Y_n|X_n),$$

which are in $\mathbb{R}$. This only makes sense however if the observed datapoints we are plugging in are iid. Thus let us assume furthermore that $Y_i|X_i$ are independent and $Y_i|X_i \sim \mathcal{N}(\mathbb{E}(Y_i|X_i), \sigma^2)$.
Defining $\tilde{Y}|X := Y|X - \sum_k a_k h_k(X)$ we see directly that $\mathbb{E}(\tilde{Y}|X) = \mathbb{E}(Y|X) - \mathbb{E}(\sum_k a_k h_k(X)) = 0$. And so by the assumption of $Y_i|X_i$ being independent and normally distributed, we get that $\tilde{Y}_i|X_i$ are iid with $\tilde{Y}_i|X_i \sim \mathcal{N}(0, \sigma^2)$. So if we plug in the sample distribution into

$$\mathbb{E}\left(\left(Y - \sum_k a_k h_k(X)\right)^2\right) = E((\tilde{Y}|X)^2)$$

we obtain the so called least squares problem:

$$\min_{a_k} \sum_i \left(y_i - \sum_k a_k h_k(\mathbf{x}_i)\right)^2, \tag{2.2}$$

which will give us - under appropriate assumptions - an estimator for the parameters a_k.

Remark 1. Without the assumption of normality we would have had the problem that we could not just plug in the sample distribution. If we can only assume $Y_i|X_i$ to be independent there is no guarantee that $\tilde{Y}_i|X_i := Y_i - \mathbb{E}(Y_i|X_i)$ is also identical distributed. This works in the case for normal data because the conditional mean determines the whole distribution - for fixed variance.
We will see how to treat the case when $Y_i|X_i$ cannot be assumed to be normal in the section about GLMs, which are a remedy for this problem.

Remark 2. We also needed to assume constant variance as otherwise, if we have $Y_i|X_i$ independent and $Y_i|X_i \sim \mathcal{N}(\mathbb{E}(Y_i|X_i), \mathbb{V}\text{ar}(Y_i|X_i))$ we couldn't use the argument that $\tilde{Y}_i|X_i$ were iid. We will see that this case is also covered by GLMs.

2.1 Additive models

In this section we will introduce additive models which are a remedy for the first problem presented in the introductory chapter.

Let us assume from now on until the section about GLMs that all $Y_i|X_i$ are independent and satisfy $Y_i|X_i \sim \mathcal{N}(\mathbb{E}(Y_i|X_i), \sigma^2)$ - so that we can follow the same chain of arguments as in the introduction of this chapter to arrive at the least squares problem (2.2).

Let us quickly talk about solvability of the least squares problem

$$\min_{a_k} \sum_{i=1}^{n} \big(y_i - \sum_{k=1}^{m} a_k h_k(\mathbf{x}_i)\big)^2, \tag{2.3}$$

with $m < n$, where $\mathbf{x}_i = (x_{i1}, ..., x_{ip})'$ is the i-th datapoint, and see how we could solve such a minimization problem in principle.

It is easy to see that the sum

$$\sum_{i} \big(y_i - \sum_{k} a_k h_k(\mathbf{x}_i)\big)^2 \tag{2.4}$$

is differentiable in its parameters a_k and calculating the partial derivative in the direction of a_j gives:

$$\frac{\partial}{\partial a_j} \sum_{i} \big(y_i - \sum_{k} a_k h_k(\mathbf{x}_i)\big)^2 = \sum_{i} \frac{\partial}{\partial a_j} \big(y_i - \sum_{k} a_k h_k(\mathbf{x}_i)\big)^2 \tag{2.5}$$

$$= -\sum_{i} 2\big(y_i - \sum_{k} a_k h_k(\mathbf{x}_i)\big) h_j(\mathbf{x}_i) \tag{2.6}$$

$$= -2\sum_{i} y_i h_j(\mathbf{x}_i) + 2\sum_{i} h_j(\mathbf{x}_i) \sum_{k} a_k h_k(\mathbf{x}_i). \tag{2.7}$$

Defining the matrix

$$\mathbf{X} := (h_j(\mathbf{x}_i))_{:,:} = \begin{pmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_m(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_n) & \cdots & \cdots & h_m(\mathbf{x}_n) \end{pmatrix} \in \mathbb{R}^{n \times m}$$

and the vectors

$$\mathbf{y} := (y_1, ..., y_n)', \mathbf{a} := (a_1, ..., a_m)',$$

we get that (2.7) is equal to

$$-2\big(\mathbf{y}'\mathbf{X}_{:,j} + \mathbf{a}'\mathbf{X}'\mathbf{X}_{:,j}\big).$$

This means that the gradient is equal to

$$-2\big(\mathbf{y}'\mathbf{X} + \mathbf{a}'\mathbf{X}'\mathbf{X}\big)' = -2\big(\mathbf{X}'\mathbf{y} + \mathbf{X}'\mathbf{X}\mathbf{a}\big). \tag{2.8}$$

A necessary condition for a minimum is that the gradient is equal to zero, so we get the so called normal equations:

$$\mathbf{X}'\mathbf{X}\mathbf{a} = \mathbf{X}'\mathbf{y}. \tag{2.9}$$

To calculate the Hessian matrix we just need to calculate the Jacobian of (2.8), which gives us:

$$\frac{\partial}{\partial \mathbf{a}} - 2\big(\mathbf{X}'\mathbf{y} + \mathbf{X}'\mathbf{X}\mathbf{a}\big) = -2\frac{\partial}{\partial \mathbf{a}}\mathbf{X}'\mathbf{X}\mathbf{a} = -2\mathbf{X}'\mathbf{X}. \tag{2.10}$$

Now if $\mathbf{X}'\mathbf{X}$ is positive definite, then, on the one hand, the normal equations (2.9) have a unique solution, namely $\mathbf{a} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, and, on the other hand, this solution is also the unique minimizer of problem (2.3) - as the Hessian is thus negative definite, meaning that our function is strictly concave.

It still remains to clarify in which case the matrix $\mathbf{X}'\mathbf{X}$ is positive definite. We have the following equivalences

$$\begin{aligned} \mathbf{X}'\mathbf{X} \text{ p.d} &\iff \mathbf{c}'\mathbf{X}'\mathbf{X}\mathbf{c} > 0 \ \forall \mathbf{c} \in \mathbb{R}^m \setminus \{\mathbf{0}\} \\ &\iff (\mathbf{X}\mathbf{c})'(\mathbf{X}\mathbf{c}) > 0 \ \forall \mathbf{c} \in \mathbb{R}^m \setminus \{\mathbf{0}\}. \end{aligned}$$

The last expression is true if and only if $(\mathbf{X}\mathbf{c})$ spans the whole space $\mathbb{R}^m$, which in return is equivalent to $\mathbf{X}$ having full column rank - implying $m \leq n$.

All in all we have therefore the following

Theorem 2.1. *The least squares problem*

$$\min_{a_k} \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{m} a_k h_k(\mathbf{x}_i) \right)^2$$

has a unique solution if $\mathbf{X}$ *has full column rank.*

Remark 3. Checking if $\mathbf{X}$ has full column rank can be done by computing the QR decomposition. There exists an orthogonal $n \times n$ matrix $\mathbf{Q}$ and an upper triangular $m \times m$ matrix $\mathbf{R}$ such that

$$\mathbf{X} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}. \tag{2.11}$$

The column rank of $\mathbf{R}$ is then the column rank of $\mathbf{X}$, as $\mathbf{Q}$ is orthogonal.

Remark 4. In reality it is not desirable to solve the normal equations, basically due to the fact that $\mathbf{X}$ can have a bad conditional number and thus solving any kind of equations which include $\mathbf{X}$ tend to deliver bad results. Instead we go back to the original problem (2.3) and use some numerical optimization scheme or, as it is described in more detail in Wood [2], we can do the following. By noticing that problem (2.3) is equivalent to minimizing $\|\mathbf{y} - \mathbf{X}\mathbf{a}\|^2$ and using the QR decomposition of $\mathbf{X}$, as well as the fact that orthogonal matrices are norm invariant, we can write

$$\begin{aligned} \|\mathbf{y} - \mathbf{X}\mathbf{a}\|^2 &= \|\mathbf{Q}'\mathbf{y} - \mathbf{Q}'\mathbf{X}\mathbf{a}\|^2 \\ &= \left\| \mathbf{Q}'\mathbf{y} - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{a} \right\|^2 = \|(\mathbf{Q}'\mathbf{y})_{1:m,:} - \mathbf{R}\mathbf{a}\|^2 + \|(\mathbf{Q}'\mathbf{y})_{m+1:n,:}\|^2 . \end{aligned}$$

We see that the solution is directly given by solving $\mathbf{R}\mathbf{a} = (\mathbf{Q}'\mathbf{y})_{1:m,:}$; which is easy to solve by back-substitution as $\mathbf{R}$ is an upper triangular matrix.

Let us now consider the following basis functions, picking up ideas from Chapter 1, $h_{\boldsymbol{\alpha}}(x) = \boldsymbol{x}^{\boldsymbol{\alpha}}$. As already mentioned in the first chapter, choosing such a basis will give us $\binom{p+d}{d}$ parameters to estimate. Even if we use the methods described in the remark to solve this problem, the number of operations to carry out would explode. This puts a limitation to what we can actually do - due to limited computational power. Not to mention that with a very large number of basis functions we also tend to run into overfitting problems.

However, it is, in many cases, also important to use higher order terms in order to get good results.

A compromise between not too many basis functions but still enough to capture higher order effects could be to leave out, or, loosely speaking, introduce interaction terms in an ordered manner. As the interaction of r variables brings $\binom{d+r}{r} - (dr+1)$ interaction terms into play, it might be beneficial to leave out higher order interactions and start from low interaction terms. For example, if $d = 10$ and $p = 5$, then the number of interaction terms of x_1 and x_2 is $\binom{10+2}{2} - (20+1) = 45$. Similarly for the interaction of x_1, x_2, x_3 we have $\binom{10+3}{3} - (30+1) = 255$ terms.

Going back to the beginning of Chapter 2 and applying what we have said so far, we could take the approach to look for functions minimizing (2.1) that have the form:

$$\begin{aligned} f(x_1, ..., x_p) &= \sum_k f_k(x_k) + \sum_{k,l} f_{k,l}(x_k, x_l) + \sum_{k,l,m} f_{k,l,m}(x_k, x_l, x_m) + ... \\ &= f_1(x_1) + f_2(x_2) + ... + f_p(x_p) + f_{12}(x_1, x_2) + f_{13}(x_1, x_3) + ... \end{aligned}$$

Lastly we want to remark that there is one problem left, namely, if we have higher order interactions like $f_{12}(x_1, x_2)$ there are identifiability problems. We could just add any function $x_1 \mapsto h(x_1)$ to f_{12} and subtract it from f_1 without changing f. So there is also a need to impose some constraints to this form to make it identifiable. We will see this later.

2.1.1 A Hilbert space approach and the backfitting algorithm

This section closely follows the book of Hastie and Tibshirani [1].

Assume that there is some reason to believe that $\mathbb{E}(Y|X)$ can be approximated by $f_1(X_1) + ... + f_p(X_p)$; so, for the moment we do not consider any interaction terms. Going back to the optimization problem (2.1), assuming $Y \in L^2(\Omega)$ with $\mathbb{E}(Y) = 0$, let us choose $\mathcal{H} \subset L^2(\Omega)$ as the space spanned by the closed spaces $\mathcal{H}_j$, for $j = 1, ..., p$, where $\mathcal{H}_j$ is the space of measurable functions f_j for which $f_j(X_j)$ is in $L^2(\Omega)$ and for which $\mathbb{E}(f_j(X_j)) = 0$ holds.

As the set of functions $\mathcal{B} := \{f(X) | f \in \mathcal{H}\}$ is closed - under some additional conditions, see [1] - therefore being a closed linear subspace of $L^2(\Omega)$, the

problem

$$\min_{f \in \mathcal{B}} \mathbb{E}\left(\left(Y - f(X)\right)^2\right) \tag{2.12}$$

can also be seen as finding the best approximation of Y, in the L^2 norm, by an element in $\mathcal{B}$.

From Hilbert space theory it is known that there is a unique minimizing element $h(X)$ and that it is given by the projection of Y onto $\mathcal{B}$.
The latter is equivalent to saying that $Y - h(X)$ is orthogonal to $\mathcal{B}$. Furthermore we get:

$$\begin{aligned} Y - h(X) \perp \mathcal{B} &\iff Y - h(X) \perp \{f_1(X_1) + ... + f_p(X_p) | f_j \in \mathcal{H}_j\} \\ &\iff Y - h(X) \perp f_j(X_j)\ \forall f_j \in \mathcal{H}_j \\ &\iff \mathbb{E}\left(\left(Y - h(X)\right) f_j(X_j)\right) = 0\ \forall f_j \in \mathcal{H}_j. \end{aligned}$$

As for any measurable set $A \in \mathcal{A}$ the function $\mathbf{1}_A(\cdot)$ also belongs to $\mathcal{H}_j$, we get $\mathbb{E}\left(\left(Y - h(X)\right)\mathbf{1}_A(X_j)\right) = 0\ \forall A \in \mathcal{A}$. This is however equivalent to $\mathbb{E}\left(\left(Y - h(X)\right)|X_j\right) = 0$ for all j.

So, all in all, as h is an element of $\mathcal{B}$, we get:

$$\mathbb{E}\left(\left(Y - \sum_{k \neq j} f_k(X_k)\right)\middle| X_j\right) = f_j(X_j), \tag{2.13}$$

for all j.

These Equations (2.13) are linear equations in a Hilbert space. They are usually impossible to solve; also because we have no knowledge about the distribution for any X_j. However these equations serve as a motivation for the following. We could try to replace the operator $\mathbb{E}\left(\cdot|X_j\right)$ and the function $f_j(X_j)$ by an approximation.

In Chapter 2 we have seen how a finite approximation to $\mathbb{E}\left(\cdot\,|X_j\right)$ could be constructed. Basically, we can construct a linear map

$$\begin{aligned} S_j(\cdot)(x_{1j}, ..., x_{nj}) : \mathbb{R}^n &\to span\{h_1, ..., h_m\} \\ \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} &\mapsto \sum_k a_k h_k(\cdot), \end{aligned}$$

by solving the least squares problem (2.2) for different $(y_1, ..., y_n)'$, through the normal equations (2.9).

From now on we will however only need that $S_j(\cdot)(x_{1j}, ..., x_{nj})$ is a linear function taking elements from $\mathbb{R}^n$ and giving back as an output a function on the same domain as x_j is defined. Furthermore we will omit writing the dependence of S_j on $(x_{1j}, ..., x_{nj})$ explicitly, for better readability.

If we replace in (2.13) now all the terms by their approximation through S_j and look at its data version, we would end up with:

$$S_j\left(\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} - \sum\nolimits_{k\neq j} \begin{pmatrix} f_k(x_{1k}) \\ \vdots \\ f_k(x_{nk}) \end{pmatrix}\right)(x_{ij}) = f_j(x_{ij}) \quad \forall i, j\,. \tag{2.14}$$

Defining $\mathbf{f}_k := (f_k(x_{1k}), ..., f_k(x_{nk}))'$, $\mathbf{y} = (y_1, ..., y_n)'$ and $\mathbf{S}_j$ as the matrix induced by

$$\begin{pmatrix} S_j(\cdot)(x_{1j}) \\ \vdots \\ S_j(\cdot)(x_{nj}) \end{pmatrix},$$

we get that (2.14) is equivalent to the following system:

$$\begin{pmatrix} \mathbf{I} & \mathbf{S}_1 & \mathbf{S}_1 & \cdots \\ \mathbf{S}_2 & \mathbf{I} & \mathbf{S}_2 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_p & \cdots & \mathbf{S}_p & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_p \end{pmatrix} = \begin{pmatrix} \mathbf{S}_1\mathbf{y} \\ \mathbf{S}_2\mathbf{y} \\ \vdots \\ \mathbf{S}_p\mathbf{y} \end{pmatrix} \in \mathbb{R}^{pn}. \tag{2.15}$$

Remark 5. Note that a vital ingredient is that we replaced $\mathbb{E}\left(\cdot\,|X_j\right)$ with a finite dimensional approximation which is linear in $\mathbf{y}$. If it were not linear, going from (2.14) to (2.15) would not be possible and would make things a lot harder.

As it is mentioned in [1], this system is usually too large to solve by inversion of the matrix, as it has np rows and np columns, and thus, takes a lot of computation time - $\mathcal{O}((np)^3)$. Therefore one goes back and uses (2.14) as a motivation for the so called backfitting algorithm, which turns out to be a Gauss-Seidel type method for solving (2.15) - in $\mathcal{O}(n)$ steps.

All in all, we get the following algorithm:

Algorithm 1 Backfitting algorithm

(1) Initialize $\mathbf{f}_j = 0$ for all $j = 1, ..., p$ and set $\hat{y}_i := y_i - \frac{1}{n}\sum_i y_i$

(2) Until the $\mathbf{f}_j$ do not change much do

For $j = 1, , ..., p$

For $i = 1, ..., n$

$$(\mathbf{f}_j)_i := S_j\Big(\hat{\mathbf{y}} - \sum_{k\neq j}\mathbf{f}_k\Big)(x_{ij})$$

end

$(\mathbf{f}_j)_i := (\mathbf{f}_j)_i - \frac{1}{n}\sum_i(\mathbf{f}_j)_i$

end

(3) The functions $f_1, .., .f_p$ are then given by

$$f_j(\cdot) := S_j\Big(\hat{\mathbf{y}} - \sum_{k\neq j}\mathbf{f}_k\Big)(\cdot)$$

and the model for Y is given by

$$\frac{1}{n}\sum_i y_i + f_1(x_1) + ... + f_p(x_p).$$

We will stop here with this approach to additive models, refer to [1] for a more complete view on AMs and their solvability, and end with the following:

Remark 6. If $\mathbf{S}_j$ are symmetric matrices, with eigenvalues in $[0, 1]$, and if the space

$$\Big\{(\mathbf{u}_1, ..., \mathbf{u}_p)'\Big|\mathbf{S}_j\mathbf{u}_j = \mathbf{u}_j\,\forall j\ \sum_{i,j}(\mathbf{u}_j)_i = 0\Big\}$$

is empty, then it can be deduced that the backfitting algorithm converges to the unique solution of (2.15), see [1].

2.1.2 Smoothing splines and a penalized approach to AMs

The approach taken in Subsection 2.1.1 is very general, meaning that we have not assumed any concrete form of the operators S_j. In this subsection we will look at a very specific way of constructing S_j. Many of the ideas and concepts presented in this subsection can be found in greater detail in [2].

As mentioned in Subsection 2.1.1, the operators S_j serve as an approximation to $\mathbb{E}(\cdot|X)$. At the beginning of Chapter 2 we have seen one possibility to construct such an approximation, namely through solving the least squares problem

$$\min_{a_k} \sum_i \big(y_i - \sum_k a_k h_k(\mathbf{x}_i)\big)^2,$$

for a fixed $(y_1, .., y_n)'$, and then setting $S_j(\cdot) := \sum_k a_k h_k(\cdot)$ - where the a_k depend on $(y_1, .., y_n)'$.
We now take a closer look into the choice of basis functions $h_1, ..., h_m$ that can be made.

Smoothing splines

Let us assume that we have observations $(y_1, x_1), ..., (y_n, x_n)$ with $x_i \in \mathbb{R}$. If we want to fit a function f to the data, one reasonable requirement usually is that the function f should be at least continuous.
We could of course use a polynomial in x to fit the data, therefore taking $h_k(x) = x^k$. However polynomials can sometimes be problematic. Trying to fit a polynomial to the data requires us to fix the degree of the polynomial. If we take a degree which is too small we might end up with a model that is too inflexible and does not predict new data too well. However, taking a degree which is too big, might lead to overfitting - meaning that the fitted function adjusts to the current data too well, usually resulting in a very wiggly function - and thus we again obtain a bad model. For example, if we generate data from the polynomial $-x + 4x^3 + \epsilon$, with $\epsilon \sim \mathcal{N}(0, (0.2)^2)$, and we tried to fit a polynomial of degree 15 to the data, we can see from Figure 2.1 that this leads to a very wiggly function - as the fitted function adapts too much to the data at hand.
A possibility would be to try to fit a polynomial with a very high degree and avoid overfitting by one of the various methods like LASSO - which basically help us decide which basis functions we should keep.
However the main problem stems from the fact that the basis functions x^k are not very local. Heuristically speaking, if a polynomial of degree d approximates an underlying function well, then adding some more basis functions, say m, and fitting thus a polynomial of degree $d+m$, does not necessarily lead to a polynomial that fits the underlying function equally well. The reason behind this is that each basis function contributes to the function evaluation over the whole range of x and thus the estimated parameters can change a lot. Not only does this imply that we may end up with a lot of computation

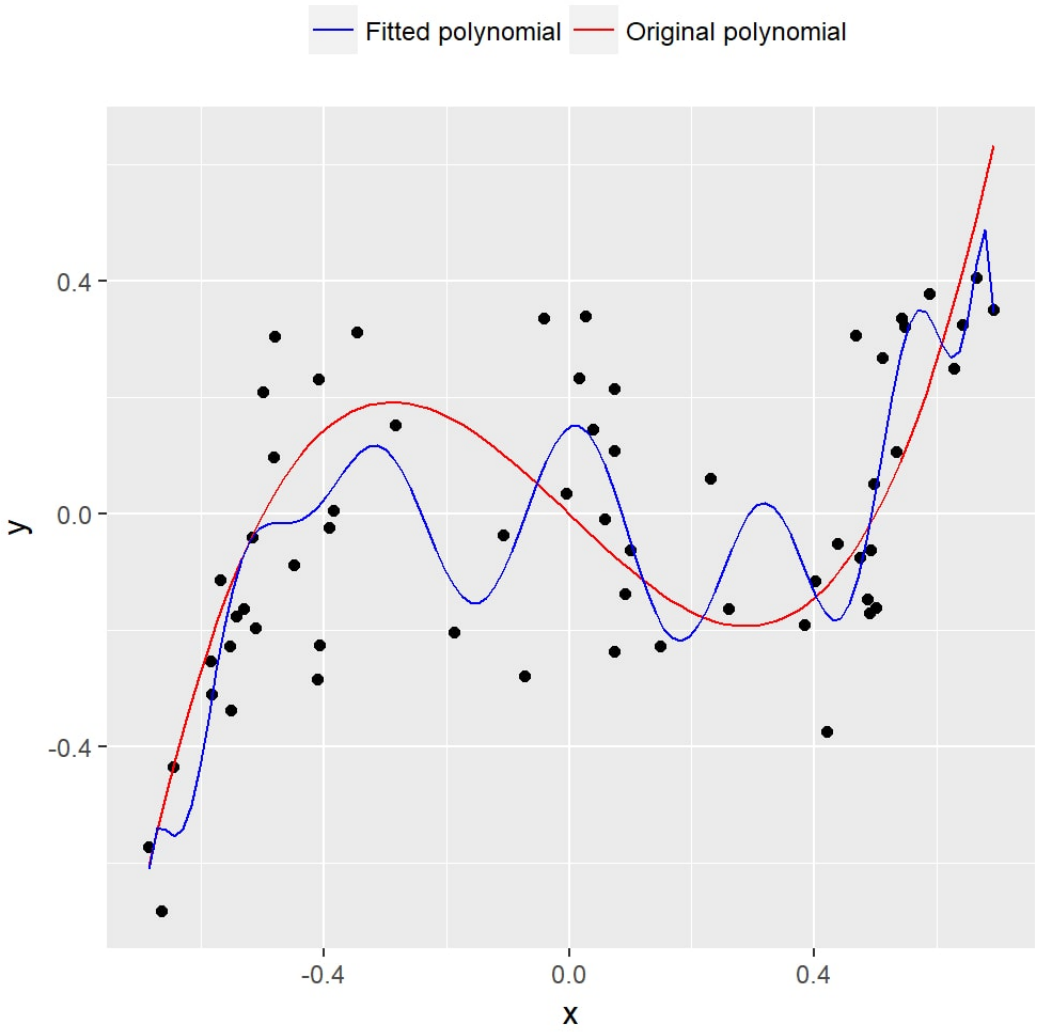


Figure 2.1: Result of overfitting

time needed when adding basis functions, but also that estimated parameters can suffer from a high variance - due to the non-locality - and so prediction becomes worse.

So one could look for basis functions h_k which behave more local. One way could be to fit a piecewise linear function or, in general, a piecewise polynomial of degree d. This means that, given fixed nodes $\zeta_1 \leq, ..., \leq \zeta_m$, where we have that all observations x_i lie in $[\zeta_1, \zeta_m]$, one fits in each interval $[\zeta_j, \zeta_{i+1}]$ a polynomial of fixed degree d; which means that the polynomial corresponding to an interval is zero outside of the latter, thus being more local. However, we still have the problem that overfitting can result in each interval in case that d is high. If we choose the nodes in such a way that not too many observations are present in each interval, low degree polynomials are sufficient to get good approximations and therefore overall wiggliness is prevented. Nevertheless, this method still has the drawback that we need to choose the degree of the piecewise polynomials and now also the nodes - even though one could for example just take quantiles.

With the goal in mind of preventing wiggliness, a nice approach to avoid having to choose the degree and, surprisingly, the nodes, is the following. Ideally we are looking for functions which are not too wiggly inside of the intervals. To prevent wiggliness we could for example try to penalize high absolute values of the second derivative of the function we are looking for; as the least wiggly function we could hope for would have $f''(x) = 0$ - hence the straight line. As this needs to be done over the whole range of x it is natural to take the integral and rather the square of the second derivative; mainly because the absolute value is a non-smooth function. So instead of minimizing the usual least squares objective (2.2) we try to minimize:

$$\sum_i \big(y_i - f(x_i)\big)^2 + \lambda \int (f''(x))^2 dx, \tag{2.16}$$

where $\lambda > 0$ is for the moment fixed, controlling the amount of penalization.

Surprisingly, it turns out that the function minimizing this functional, over all functions for which the term $\int (f''(x))^2 dx$ makes sense, is a so called natural cubic spline with nodes in x_i, which is a special type of piecewise polynomial of degree three - described below.

This can easily be seen by the following fact. For a fixed function f, we have that any natural cubic spline p, interpolating f in $x_1, ..., x_n$, fulfills

$\int (p''(x))^2 dx \leq \int (f''(x))^2 dx$, see [2]. Hence the minimizer of (2.16) must also be a cubic spline, because if f minimizes the latter, then by choosing a piecewise polynomial with nodes x_i, which exactly interpolates f in x_i, we can basically replace f in (2.16) with such a piecewise polynomial. The first part of the objective function thus stays unchanged and the second part is at most as high as before.

Now as mentioned above, a natural cubic spline is a special type of piecewise polynomial of degree three. First of all, an M-spline is commonly defined as a piecewise polynomial, where the degree of each polynomial in each of the $m-1$ intervals $[\zeta_i, \zeta_{i+1}]$ is $M-1$, with the addition of having continuous derivative up to order $M-2$ in (ζ_1, ζ_m). Thus, for an M-spline we need to estimate $M(m-1)-(M-1)(m-2) = M+m-2$ parameters. A basis for such an M-spline could explicitly be given by

$$\begin{aligned} h_k(x) &:= x^{k-1} && \text{for } k = 1, ..., M \\ h_i(x) &:= \max(0, x-\zeta_i)^{M-1} && \text{for } i = 2, ..., m-1. \end{aligned}$$

Actually in practice it would be more advisable to use so called $B-$splines or $P-$splines which are numerically much more stable, see [2].
A natural cubic spline is now an $M = 4$ spline with the addition of two more restrictions to the spline, namely $f''(x_1) = f''(x_m) = 0$ - meaning that f is linear outside of $[\zeta_1, \zeta_m]$. Thus we have $M+m-2-2 = 4+m-2-2 = m$ parameters to estimate now.

Remark 7. So far we have only looked at splines in one variable. However, there is also the possibility of taking more than one variable, say $x_1, ..., x_p$. In this case we could try to minimize a similar objective function to (2.16), where the second term is just replaced by an integral penalizing the wiggliness of the "surfaces" instead of the curves.
Popular choices are for example

$$\sum_{\eta_1+\cdots+\eta_p=m} \lambda \frac{m!}{\eta_1 \dots \eta_p} \int \left(\frac{\partial^m f}{\partial x_1^{\eta_1} \dots \partial x_p^{\eta_p}} \right)^2 dx_1 \dots dx_p \tag{2.17}$$

$$\sum_k \lambda_k \int \left(\frac{\partial^2}{\partial x_k^2} f(x_1, ..., x_p) \right)^2 dx_k. \tag{2.18}$$

The motivation behind these is the following. Intuitively we would like for a function to have the property that in each direction the second derivative is close to zero, leading to the condition that the sum of all the second

derivatives is close to zero. This approach would be incorporated in (2.17) by choosing $m = 2$.
However, the problem with this penalty is that we need the condition $2m > p + 1$ to hold, to be able to obtain nice closed form solutions, see [2]. But, for example using $p = 8$ and so $m = 5 > \frac{8}{2} = 4$, we get that the penalty minimizes 5th-order derivatives, which is not very intuitive as to if this minimizes the wiggliness - as this means that we look to obtain solutions which behave like quadratic polynomials. Also this approach is computationally very expensive and so usually the second penalty (2.18) is preferable.
As the penalty (2.18) is somewhat suited for the penalized approach of additive models as we will soon see, we will stop here and instead refer to [2].

A penalized approach to AMs

Let us return now to the original problem where we tried to minimize (2.1) under the assumption that f has an additive structure, that is $f_1(X_1) + \cdots + f_p(X_p)$. At the end of the introduction of Chapter 2 we had basically replaced the probability measure in $\mathbb{E}\left(\left(Y - f(X)\right)^2\right)$ by the sample probability measure. In this spirit, if we wanted to impose an additive structure, we would get the problem:

$$\min_{f_1,\dots,f_p} \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{p} f_k(x_{ik})\right)^2. \tag{2.19}$$

Of course, as it can be easily seen, we need to fix the space in which the functions f_k lie, as otherwise taking $f_1(x_{i1}) := y_i$ would give a perfect fit.
As discussed before, one desirable property is to reduce wiggliness of the function $f_1(x_1) + \cdots + f_p(x_p)$.
Due to the additive nature of this function it seems appropriate to use a penalty term of the form (2.18). So end up with the minimization problem:

$$\min_{f_1,\dots,f_p} \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{p} f_k(x_{ik})\right)^2 + \sum_{k=1}^{p} \lambda_k \int \left(\frac{\partial^2}{\partial x_k^2} f_k(x_k)\right)^2 dx_k, \tag{2.20}$$

where the functions f_i lie in the function space such that the integral terms make sense - typically a Sobolev space - and $\lambda_k > 0$ are fixed for the moment.

As the penalty and the function f we are looking for is of an additive nature, it is easy to see, by the same arguments than before, that the functions f_i which minimize this problem are all natural cubic splines.

Because of this we can write $f_k(\cdot) = \sum_{j=1}^{n} a_{jk} h_{jk}(\cdot)$, for appropriate h_{jk}. This means that (2.20), when using the definitions

$$\begin{aligned}
&\mathbf{X}^k := (h_{jk}(x_{ik}))_{i,j} && \in \mathbb{R}^{n\times n}\\
&\mathbf{X} := (\mathbf{X}^k_{i,:})_{i,k} = \begin{pmatrix} \mathbf{X}^1_{1,:} & \cdots & \mathbf{X}^p_{1,:} \\ \vdots & \ddots & \vdots \\ \mathbf{X}^1_{n,:} & \cdots & \mathbf{X}^p_{n,:} \end{pmatrix} && \in \mathbb{R}^{n\times n\cdot p}\\
&\mathbf{a}^k := (a_{1k}, ..., a_{nk})' && \in \mathbb{R}^{n\times 1}\\
&\mathbf{a} := (\mathbf{a}^1, ..., \mathbf{a}^p)' && \in \mathbb{R}^{n\cdot p\times 1}\\
&\mathbf{S}^k := \left(\int \left(\frac{\partial^2}{\partial x_k^2} (h_{jk}(x_k) h_{j'k}(x_k)) \right)^2 dx_k \right)_{j,j'} && \in \mathbb{R}^{n\times n}\\
&\mathbf{S} := \begin{pmatrix} \lambda_1 \mathbf{S}^1 & 0 & \cdots \\ 0 & \ddots & 0 \\ \vdots & 0 & \lambda_p \mathbf{S}^p \end{pmatrix} && \in \mathbb{R}^{n\cdot p\times n\cdot p},
\end{aligned}$$

is equivalent to:

$$\begin{aligned}
&\min_{a_{jk}} \sum_i \left(y_i - \sum_k \sum_j a_{jk} h_{jk}(x_{ik})\right)^2 + \sum_k \lambda_k \int \left(\frac{\partial^2}{\partial x_k^2} \sum_j a_{jk} h_{jk}(x_k) \right)^2 dx_k \\
&= \min_{a_{jk}} \sum_i \left(y_i - \sum_k \mathbf{X}^k_{i,:} \mathbf{a}^k\right)^2 + \\
&\qquad \sum_k \sum_j \sum_{j'} a_{jk} a_{j'k} \lambda_k \int \left(\frac{\partial^2}{\partial x_k^2} (h_{jk}(x_k) h_{j'k}(x_k)) \right)^2 dx_k \\
&= \min_{\mathbf{a}^k} \sum_i \left(y_i - (\mathbf{X}^1_{i,:}, ..., \mathbf{X}^p_{i,:})\mathbf{a}\right)^2 + \sum_k \lambda_k (\mathbf{a}^k)' \mathbf{S}^k \mathbf{a}^k \\
&= \min_{\mathbf{a}} \|\mathbf{y} - \mathbf{X}\mathbf{a}\|^2 + \mathbf{a}'\mathbf{S}\mathbf{a}.
\end{aligned}$$

It is not hard to show, by calculating the gradient, that the solution to this problem is given by

$$\mathbf{a} = (\mathbf{X}'\mathbf{X} + \mathbf{S})^{-1} \mathbf{X}'\mathbf{y}, \tag{2.21}$$

provided that $\mathbf{X}'\mathbf{X} + \mathbf{S}$ is positive definite (p.d). As $\mathbf{S}$ is made up of $\mathbf{S}^k$, which are positive semi-definite, because $(\mathbf{a}^k)\mathbf{S}\mathbf{a}^k$ is an integral over a positive function, we get that $\mathbf{S}$ is always positive semi-definite.
Checking if $\mathbf{X}'\mathbf{X} + \mathbf{S}$ is p.d can be done by using the QR decomposition, see

(2.11).

Note that we cannot really count on proofing positive definiteness of $\mathbf{X}'\mathbf{X}+\mathbf{S}$ by showing that $\mathbf{X}$ has full column rank - as this is a $n \times np$ matrix.

Unfortunately, we can readily see that we have multiple solutions. A solution surely exists, because in the case that $\mathbf{X}$ has rank higher or equal to one at least one solution exists - as the objective function is then convex and goes in this case to plus infinity for $||\mathbf{a}|| \to \infty$ - and in the case that it has rank zero, then the solution is given by $\mathbf{a} = \mathbf{0}$. Multiple solutions thus exist, as adding a constant to any function f_{k_0} and subtracting it from another function f_{k_1} leaves f unchanged.

Let us just mention that it is therefore better to look at the slightly changed model where we set all but one coefficient a_{jk}, which correspond to $h_{jk} \equiv 1$, to zero to obtain

$$\min_{\tilde{\mathbf{a}}} ||\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{a}}||^2 + \tilde{\mathbf{a}}'\tilde{\mathbf{S}}\tilde{\mathbf{a}}. \tag{2.22}$$

We will discuss this in more detail in Chapter 3.

We have the following

Corollary 2.1. If $\tilde{\mathbf{X}}'\tilde{\mathbf{X}} + \tilde{\mathbf{S}}$ is not p.d then multiple solutions can exist. If $\tilde{\mathbf{X}}'\tilde{\mathbf{X}} + \tilde{\mathbf{S}}$ is p.d, then Problem (2.22) is solved by

$$\mathbf{a} = (\tilde{\mathbf{X}}'\tilde{\mathbf{X}} + \tilde{\mathbf{S}})^{-1}\tilde{\mathbf{X}}'\mathbf{y}.$$

and all f_k are unique.

We will end this subsection by the following three remarks.

Remark 8. In the case that $\tilde{\mathbf{X}}$ does not have full column rank it can be shown that the functions f_k are not unique, see [4].

Remark 9. It seems as if the approach described in this subsection is inferior to the one described in the Subsection 2.1.1. By not having to use specific operators S_j in 2.1.1 there is a certain liberty in choosing them. We only need to have the conditions fulfilled, which are mentioned at the end of Subsection 2.1.1, so that backfitting converges to a unique solution. As is pointed out in [1], one of the many choices could be natural cubic splines, regression splines, polynomials or surface smoothers, guaranteeing convergence.

If we work with continuous variables $x_1, ..., x_p$ for which we can expect $f_1, ..., f_p$ to be smooth functions, smoothing splines seem a reasonable choice, as explained in this subsection. In this case the two approaches are the same. If, for all j, we choose:

$$S_j : \mathbb{R}^n \to \{\text{space of natural cubic splines in } (x_{1j}, ..., x_{nj})\}$$
$$S_j(\tilde{y_1}, ..., \tilde{y_n}) := \arg\min_{f_j} \sum_i \left(\tilde{y_i} - f_j(x_{ij})\right)^2 + \lambda_j \int \left(\frac{\partial^2}{\partial x_j^2} f_j(x_j)\right)^2 dx_j,$$

we get that the step in the backfitting algorithm

$$(\mathbf{f}_j)_i := S_j\Big(\tilde{\mathbf{y}} - \sum_{k \neq j} \mathbf{f}_k\Big)(x_{ij})$$

is equivalent to finding the function f_j which minimizes

$$\sum_i \left(\tilde{y_i} - f_j(x_{ij}))\right)^2 + \lambda_j \int \left(\frac{\partial^2}{\partial x_j^2} f_j(x_j)\right)^2 dx_j,$$

where

$$\tilde{y_i} := \Big(y_i - \sum_{k \neq j} f_k(x_{ik})\Big).$$

This means that backfitting is the same as solving the minimization problem (2.20) stepwise in $f_1, ..., f_p$ - thus for this choice of S_j we get the same results. In Chapter 3 we will only consider the approach taken in this subsection, as GAMs - and especially robust GAMs - are much more accessible in this way. If we do not expect the f_k to be smooth functions then the operator approach to AMs may work better - if for example we consider only piecewise constant functions, we want to use KNN smoothers or regression trees.

Remark 10. Lastly, we want to remark that with the methods so far it is also possible to take into account functions of interacting variables, e.g. $f(X_1, X_3)$, or even projections of such, e.g. $f(a_1X_1 + a_3X_3)$ - where the latter leads to projection pursuit regression. If for example we would like to model $\mathbb{E}(Y|X_1, ..., X_p)$ as a function $f(X_1) + f(X_2) + f(X_3, ..., X_p)$ we only need to take, in the operator approach, operators S_1, S_2, S_3 associated with $\tilde{X}_1 := X_1, \tilde{X}_2 := X_2, \tilde{X}_3 := (X_3, ..., X_p)$. So we could, for example, take S_3 as the smoothing operator defined through the minimization problem

$$\min_f \sum_i \left(y_i - f(\tilde{x}_{i3})\right)^2 + \text{ penalty (2.17) or (2.18)}$$

and S_1 as well as S_2 as the operators defined through smoothing splines. It is just as easy to do the same in the approach described in this section, where we would have to replace the penalty by a suitable penalty for more than one variable.
If some of the variables are categorical ones, both approaches work in a similar manner. We could just as well model each function also depending on categorical variables and then choose suitable operators or penalties. We will see this in the chapter about GAMs.

2.2 Generalized linear models

The following section closely follows the book by Wood [2].

At the beginning of this chapter we considered the following problem

$$\min_{f \in \mathcal{H}} \mathbb{E}\left(\big(Y - f(X)\big)^2\right) \tag{2.23}$$

and then found that $f(X) = \mathbb{E}(Y|X)$ is solving it.

We were only able continue from there on by assuming that all $Y_i|X_i$ are independent and normally distributed, i.e. $Y_i|X_i \sim \mathcal{N}(\mathbb{E}(Y_i|X_i), \sigma^2)$, with constant variance.

If this is not the case however, we could not have arrived at the least squares problem, as it was explained at the end of the introduction of this chapter. If $Y_i|X_i$ violates the assumption of normality, it makes therefore more sense to take a likelihood approach, meaning that if the distribution at hand depends on some way on the conditional expectation, as a parameter, we could try to maximize its likelihood instead:

$$\max_{a_k} \sum_i l\Big(\mathbb{E}(y_i|\mathbf{x}_i) = \sum_{k=1}^{m} a_k h_k(\mathbf{x}_i)\Big).$$

That this approach seems reasonable is also supported by the fact that if $Y_i|X_i$ is independent and normally distributed, taking the approach (2.23) is the same as taking a maximum likelihood approach; because looking at the problem

$$\max_{a_k} \mathbb{E}\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{\|Y - \sum_k a_k h_k(X)\|^2}{2\sigma^2})\right),$$

for a fixed σ, we would have ended up with the same a_k.

There is however yet another problem with taking such a likelihood approach. Assume that $Y_i|X_i$ is Poisson distributed, i.e has the density

$$f(y_i) = \frac{\lambda(\mathbf{x}_i)^{y_i}}{y_i!} \exp(-\lambda(\mathbf{x}_i)),$$

with parameter $\lambda(\mathbf{x}_i) > 0$. For such a variable we also have $\mathbb{E}(Y_i|X_i) = \lambda(X_i)$, and thus it follows that the conditional expectation also has to be positive. If we would now naively model $\mathbb{E}(Y_i|X_i)$ by $\sum_k a_k h_k(X_i)$, as it was mentioned at the beginning of Chapter 2, we would not only run into problems if we tried to maximize

$$\max_{a_k} \prod_i \frac{\big(\sum_k a_k h_k(\mathbf{x}_i)\big)^{y_i}}{y_i!} \exp\Big(-\sum_k a_k h_k(\mathbf{x}_i)\Big)$$

- as there is no guarantee that $\sum_k a_k h_k(\mathbf{x}_i)$ stays positive during maximization - but also encounter the problem that when we try to predict Y, for given X, by $\sum_k a_k h_k(X)$, we might get a negative value.

So an idea could be to compose $f := \sum_k a_k h_k$ with another function g such that $g(f(\cdot))$ always stays positive. This would mean that, provided that we have fixed a suitable function g, we could look at the following modified problem:

$$\max_{a_k} \prod_i \frac{\big(g(\sum_k a_k h_k(\mathbf{x}_i))\big)^{y_i}}{y_i!} exp\Big(-g\Big(\sum_k a_k h_k(\mathbf{x}_i)\Big)\Big),$$

thus trying to approximate $\mathbb{E}(Y|X)$ by $g(f(X))$.

2.2.1 The exponential family

In the introduction of this section we have talked about the idea of introducing a function g in order to match the possible image of $g(f)$ with the range of possible values of $\mathbb{E}(Y|X)$. This immediately leads to the question of how to choose such a function g.
Fortunately, for the so called exponential family, which covers a wide range of different distributions, we are able to obtain, in many cases, a canonical function g. We will therefore only consider this family of distributions in the following - writing from now on Y instead of $Y|X$.

Definition 2.1. *Y is said to belong to the exponential family if its density is of the form:*

$$f_{\theta,\psi}(y) = exp\left(\frac{y\theta - b(\theta)}{a(\psi)} - h(y,\psi)\right),$$

where $\psi > 0$, b is a twice continuously differentiable function, with $b'' > 0$, and a and h be such that the log-likelihood function l is regular enough in $\theta \in \Theta \subset \mathbb{R}$.

Remark 11. For l to be regular enough means that: θ is defined on an open set, l is twice continuously differentiable in θ, Y has finite variance, the integrals $\int l\,dy$, $\int \frac{\partial}{\partial\theta} l\,dy$ and $\int \frac{\partial^2}{\partial\theta} l\,dy$ exist and are finite, and integration can be interchanged with differentiation in the latter two.

Remark 12. Using exponential families to model a random variable Y is much more than just transforming Y with some function f hoping that $f(Y)$ is normally distributed. Firstly there might not even exist such a function f to achieve this, secondly even if there was such a function it might be hard to find its form, and at last, as we will soon see, exponential families have the advantage that they allow us to model non constant variance and a relationship between expectation and variance. So in this sense GLMs are a necessary extension.

Now, as it is described in [2], it is easy for a random variable Y belonging to the exponential family to find a function g which fulfills the property we are looking for.
The log likelihood and its first and second derivative in θ is given by

$$l(\theta,\psi) = \frac{Y\theta - b(\theta)}{a(\psi)} - h(Y,\psi) \tag{2.24}$$

$$\frac{\partial}{\partial\theta} l(\theta,\psi) = \frac{Y - b'(\theta)}{a(\psi)} \tag{2.25}$$

$$\frac{\partial^2}{\partial\theta^2} l(\theta,\psi) = -\frac{b''(\theta)}{a(\psi)}. \tag{2.26}$$

From (2.25) it follows that:

$$\mathbb{E}\left(\frac{\partial}{\partial\theta} l(\theta,\psi)\right) = \mathbb{E}\left(\frac{Y - b'(\theta)}{a(\psi)}\right) = \frac{\mathbb{E}(Y) - b'(\theta)}{a(\psi)}.$$

However as

$$\begin{aligned}\mathbb{E}\left(\frac{\partial}{\partial\theta}l(\theta,\psi)\right)=\mathbb{E}\left(\frac{\partial}{\partial\theta}log(f_{\theta,\psi}(Y))\right)&=\mathbb{E}\left(\frac{\frac{\partial}{\partial\theta}f_{\theta,\psi}(Y)}{f_{\theta,\psi}(Y)}\right)\\&=\int\frac{\frac{\partial}{\partial\theta}f_{\theta,\psi}(y)}{f_{\theta,\psi}(y)}f_{\theta,\psi}(y)\,dy\\&=\int\frac{\partial}{\partial\theta}f_{\theta,\psi}(y)\,dy\\&=\frac{\partial}{\partial\theta}\int f_{\theta,\psi}(y)\,dy\\&=\frac{\partial}{\partial\theta}1=0,\end{aligned}$$

we get all in all:

$$\frac{\mathbb{E}(Y)-b'(\theta)}{a(\psi)}=0\iff\mathbb{E}(Y)=b'(\theta).\tag{2.27}$$

This means that if we choose $g\equiv b'$, we get that the image will always match the range of the possible values of $\mathbb{E}(Y)$.

Furthermore, using (2.26) we get:

$$-\frac{b''(\theta)}{a(\psi)}=\mathbb{E}\left(-\frac{b''(\theta)}{a(\psi)}\right)=\mathbb{E}\left(\frac{\partial^2}{\partial\theta^2}l(\theta,\psi)\right)=\mathbb{E}\left(\frac{\partial}{\partial\theta}\frac{\partial}{\partial\theta}l(\theta,\psi)\right),$$

where the latter is further equal to

$$\begin{aligned}&\mathbb{E}\left(\frac{\partial}{\partial\theta}\frac{\frac{\partial}{\partial\theta}f_{\theta,\psi}(Y)}{f_{\theta,\psi}(Y)}\right)\\&=\int\left(\frac{\partial}{\partial\theta}\frac{\frac{\partial}{\partial\theta}f_{\theta,\psi}(y)}{f_{\theta,\psi}(y)}\right)f_{\theta,\psi}(y)\,dy\\&=\int\frac{\partial}{\partial\theta}\left(\frac{\frac{\partial}{\partial\theta}f_{\theta,\psi}(y)}{f_{\theta,\psi}(y)}f_{\theta,\psi}(y)\right)dy-\int\frac{\frac{\partial}{\partial\theta}f_{\theta,\psi}(y)}{f_{\theta,\psi}(y)}\frac{\partial}{\partial\theta}f_{\theta,\psi}(y)\,dy\\&=\int\frac{\partial^2}{\partial\theta^2}f_{\theta,\psi}(y)\,dy-\int\frac{\frac{\partial}{\partial\theta}f_{\theta,\psi}(y)}{f_{\theta,\psi}(y)}\frac{\frac{\partial}{\partial\theta}f_{\theta,\psi}(y)}{f_{\theta,\psi}(y)}f_{\theta,\psi}(y)\,dy.\end{aligned}$$

Using the property that we can interchange derivation and integration, we

get that the last term is equal to

$$\begin{aligned}\frac{\partial^2}{\partial\theta^2}\int f_{\theta,\psi}(y)\,dy - \mathbb{E}(l(\theta,\psi)^2) &= -\,\mathbb{E}\left(\left(\frac{\partial^2}{\partial\theta^2}l(\theta,\psi)\right)^2\right)\\ &= -\,\mathbb{E}\left(\left(\frac{Y-b'(\theta)}{a(\psi)}\right)^2\right)\\ &= -\frac{Var(Y)}{a(\psi)^2},\end{aligned}$$

where in the forelast equality we used (2.25). So we finally arrive at

$$\mathbb{V}\mathrm{ar}(Y) = b''(\theta)a(\psi). \tag{2.28}$$

Remark 13. We see from Equation (2.28) one very big advantage of the exponential family approach, opposed to the case of normally distributed variables Y, with constant variance. Namely, the variance of Y is here now connected to the mean of Y and can thus change for different X.

Table (2.1) sums up some key facts about some of the most common distributions - which are part of the exponential family.

2.2.2 The maximum likelihood problem and the IRLS

Suppose that it can be assumed that $Y|X$ is Gamma distributed. As it is shown in Table (2.1) we have that the mean function is equal to $\mathbb{E}(Y) = b'(\theta) = -\frac{1}{\theta}$. This means that when we model θ, which must be negative, by $\sum_k a_k h_k$, we would still have to restrict the coefficients a_k in such a way that $-(\sum_k a_k h_k)^{-1}$ is always positive - for all possible x which we allow.

We will see later, in Chapter 3, how to deal with such a case. In the following we will only look at the very common case $a(\psi) = \psi$ and use $g \equiv b'$, where the range of θ is $\mathbb{R}$. Continuing our train of thought from before, we have the following problem

$$\max_{a_k,\psi} l\Big(\sum_{k=1}^{m} a_k h_k(X), \psi\Big),$$

with $m < n$, or, in the data version - with observations $(y_1, \mathbf{x}_1), ..., (y_n, \mathbf{x}_n)$ - and the abbreviation $\theta(\mathbf{a}, i) := \sum_{k=1}^{m} a_k h_k(\mathbf{x}_i)$:

$$\max_{a_k,\psi} \sum_i l\big(\theta(\mathbf{a}, i), \psi\big) \tag{2.29}$$

$$= \max_{a_k,\psi} \sum_i \frac{y_i\theta(\mathbf{a}, i) - b(\theta(\mathbf{a}, i))}{\psi} - h(y_i, \psi). \tag{2.30}$$

Table 2.1: Some distributions belonging to the exponential family

	Normal	Binomial	Poisson	Exponential	Gamma	Inverse Gaussian
$f(y)$	$\frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(y-\mu)^2}{2\sigma^2})$	$\binom{n}{y}p^y(1-p)^{n-y}$	$\frac{\lambda^y}{y!}\exp(-\lambda)$	$\lambda\exp(-\lambda y)$	$\frac{\beta^\alpha}{\Gamma(\alpha)}y^{\alpha-1}\exp(-\beta y)$	$\sqrt{\frac{\gamma}{2\pi y^3}}\exp(-\frac{\gamma(y-\mu)^2}{2\mu^2 y})$
Parameters	$\mu\in\mathbb{R},\sigma^2>0$	$p\in(0,1),n\in\mathbb{N}_0$	$\lambda>0$	$\lambda>0$	$\alpha>0,\beta>0$	$\gamma>0,\mu>0$
$\mathbb{E}(Y)$	μ	np	λ	$\frac{1}{\lambda}$	$\frac{\alpha}{\beta}$	μ
$\theta=$	μ	$\log(\frac{p}{1-p})$	$\log(\lambda)$	$-\lambda$	$-\frac{\beta}{\alpha}$	$-\frac{1}{2\mu^2}$
Range of y	$(-\infty,+\infty)$	$\{0,...,n\}$	$\mathbb{N}_0$	$(0,+\infty)$	$(0,+\infty)$	$(0,+\infty)$
Range of θ	$(-\infty,+\infty)$	$(-\infty,+\infty)$	$(-\infty,+\infty)$	$(-\infty,0)$	$(-\infty,0)$	$(-\infty,0)$
$\psi=$	σ^2	1	1	1	$\frac{1}{\alpha}$	$\frac{1}{\gamma}$
$a(\psi)$	ψ	1	1	1	ψ	ψ
$h(y,\psi)$	$-\big(\frac{y^2}{\psi}+\log(2\pi\psi)\big)$	$\log(\binom{n}{y})$	$-\log(y!)$	0	$\alpha\log(\alpha y)-\log(y\Gamma(\alpha))$	$-\frac{1}{2}\big(\log(2\pi y^3\psi)+\frac{1}{\psi y}\big)$
$b(\theta)$	$\frac{\theta^2}{2}$	$n\log(1+\exp(\theta))$	$\exp(\theta)$	$-\log(-\theta)$	$-\log(-\theta)$	$-\sqrt{-2\theta}$
$g(\theta)=b'(\theta)$	θ	$n\frac{\exp(\theta)}{1+\exp(\theta)}$	$\exp(\theta)$	$-\frac{1}{\theta}$	$-\frac{1}{\theta}$	$(-2\theta)^{-\frac{1}{2}}$
Range of $\mathbb{E}(Y)$	$(-\infty,+\infty)$	$(0,n)$	$(0,+\infty)$	$(0,+\infty)$	$(0,+\infty)$	$(0,+\infty)$

An approach for solving (2.30) would be to look for points where the gradient vanishes. So differentiating the likelihood in the direction of a_j, we get:

$$\begin{aligned}
&\frac{\partial}{\partial a_j}\sum_i \frac{y_i\theta(\mathbf{a},i) - b(\theta(\mathbf{a},i))}{a(\psi)} - h(y_i,\psi) \\
&= \frac{1}{\psi}\sum_i \left(y_i \frac{\partial}{\partial a_j}\theta(\mathbf{a},i) - \frac{\partial}{\partial a_j} b(\theta(\mathbf{a},i)) \right) \\
&= \frac{1}{\psi}\sum_i \left(y_i \ \frac{\partial}{\partial a_j}\theta(\mathbf{a},i) - b'(\theta(\mathbf{a},i))\frac{\partial}{\partial a_j}\theta(\mathbf{a},i) \right) \\
&= \frac{1}{\psi}\sum_i \Big(y_i - b'(\theta(\mathbf{a},i)) \Big) \frac{\partial}{\partial a_j}\theta(\mathbf{a},i).
\end{aligned}$$

As we are looking for points where the gradient vanishes, this thus means that we are looking for $\mathbf{a}$ and ψ which fulfill

$$\sum_i \Big(y_i - b'(\theta(\mathbf{a},i)) \Big) \nabla_{\mathbf{a}}\, \theta(\mathbf{a},i) = 0 \tag{2.31}$$

$$\frac{\partial}{\partial \psi}\sum_i l\big(\theta(\mathbf{a},i),\psi\big) = 0. \tag{2.32}$$

As the the first equation does not contain ψ, we could in principle first solve Equation (2.31), to obtain a solution $\mathbf{a}_0$, and then solve (2.32). However, as the second equation usually depends in a very non-linear way on ψ and might have multiple solutions, we only solve the first equation and then estimate ψ - this we will see in the next subsection.

To solve (2.31) we must resort to numerical methods, as it is in most cases impossible to solve this equation exactly. One very popular method of choice is the Newton-Raphson, because it is computationally very cheap in comparison to other methods.

Basically the Newton-Raphson method consists in linearising (2.31), meaning that for a point solving $\mathbf{a}_0$ (2.31), which is equivalent to

$$\sum_i \nabla l\big(\theta(\mathbf{a}_0,i),\psi\big) = 0,$$

we can deduce that close to this $\mathbf{a}_0$ we have:

$$\sum_i \nabla l\big(\theta(\mathbf{a},i),\psi\big) + \mathcal{H}_l\left(\theta(\mathbf{a},i),\psi\right)(\mathbf{a}_0 - \mathbf{a}) + rest = 0,$$

where $\mathcal{H}_l$ is the Hessian matrix of the likelihood.

The idea is thus to, hopefully, end up with a stepwise approach such that $\mathbf{a}_0$ can be obtained by an iterative method:

$$\mathbf{a}_{j+1} = \mathbf{a}_j - \Big(\sum_i \mathcal{H}_l \big(\theta(\mathbf{a}_j, i), \psi\big) \Big)^{-1} \sum_i \nabla l\big(\theta(\mathbf{a}_j, i), \psi\big).$$

Luckily, if (2.31) has at least one solution and if $\mathbf{X} \in \mathbb{R}^{n \times m}$, $n > m$, has full column rank, it can be shown that (2.31) has exactly one solution - and it is found by the iteration steps above. This can be seen by the fact that problem (2.30) is concave in the parameters a_k, because the second derivative of the likelihood is given by:

$$\begin{aligned}
&\frac{\partial^2}{\partial a_l \partial a_j} \sum_i l\big(\theta\big(\mathbf{a}, i\big), \psi\big) \\
&= \frac{1}{\psi} \frac{\partial}{\partial a_l} \sum_i \Big(y_i - b'(\theta(\mathbf{a}, i)) \Big) \frac{\partial}{\partial a_j} \theta(\mathbf{a}, i) \\
&= \frac{1}{\psi} \frac{\partial}{\partial a_l} \sum_i \Big(y_i - b'(\theta(\mathbf{a}, i)) \Big) h_j(\mathbf{x}_i) \\
&= \frac{1}{\psi} \sum_i \Big(- b''(\theta(\mathbf{a}, i)) h_l(x_i) \Big) h_j(\mathbf{x}_i),
\end{aligned}$$

where we used $\frac{\partial}{\partial a_j} \theta(\mathbf{a}, i) = h_j(\mathbf{x}_i)$. Thus the Hessian matrix is given as

$$-\frac{1}{\psi} \mathbf{X}' \mathbf{D} \mathbf{X},$$

where $\mathbf{D}$ is a diagonal matrix with elements $b''(\theta(\mathbf{a}, i))$.

Basically by the same reasoning as for Theorem (2.1) one can show that $\frac{1}{\psi} \mathbf{X}' \mathbf{D} \mathbf{X}$ is positive definite - b'' is strictly positive - and so the Hessian is negative definite. This means that the objective function of problem (2.30) is strictly concave, in $\mathbf{a}$, and therefore the objective function of (2.30) has a unique maximum in $\mathbf{a}$ - for any fixed ψ - if one exists.
For a strictly concave function, on $\mathbb{R}^m$, it is well known that the Newton-Raphson method always finds its unique maximum approximately.
All together we have proven the following

Theorem 2.2. *Assuming that problem (2.31) has a solution and that* $\mathbf{X}$ *has full column rank, it holds that (2.31) has a unique solution. The latter is also the unique maximum of the objective function of (2.30), in* $\mathbf{a}$*, for any* ψ*, and can approximately be found by the Newton-Raphson method*

$$\mathbf{a}_{j+1} = \mathbf{a}_j - \left(\sum_i \mathcal{H}_l \left(\theta(\mathbf{a}_j, i), \psi \right) \right)^{-1} \sum_i \nabla l\big(\theta(\mathbf{a}_j, i), \psi \big), \tag{2.33}$$

for any chosen starting value $\mathbf{a}_0$*.*

One big problem with the Newton-Raphson method is that, in our case, we need to calculate the Hessian matrix for each iteration - which can be very expensive to do. Fortunately, one step of Newton-Raphson (2.33) is equivalent to a weighted least squares problem.
As we can write

$$\begin{aligned}
\mathbf{a}_{j+1} &= \mathbf{a}_j - \left(\sum_i \mathcal{H}_l \left(\theta(\mathbf{a}_j, i), \psi \right) \right)^{-1} \sum_i \nabla l\big(\theta(\mathbf{a}_j, i), \psi \big) && (2.34)\\
&= \mathbf{a}_j + \left(\frac{1}{\psi} \mathbf{X}'\mathbf{D}\mathbf{X} \right)^{-1} \frac{1}{\psi} \Big(\mathbf{X}'\mathbf{y} - \mathbf{X}'(b'(\theta(\mathbf{a}_j, i)))_{:} \Big) && (2.35)\\
&= \Big(\mathbf{X}'\mathbf{D}\mathbf{X} \Big)^{-1} \Big(\mathbf{X}'\mathbf{D}\mathbf{X}\mathbf{a}_j + \mathbf{X}'\mathbf{y} - \mathbf{X}'(b'(\theta(\mathbf{a}, i)))_{:} \Big) && (2.36)\\
&= \Big(\mathbf{X}'\mathbf{D}\mathbf{X} \Big)^{-1} \mathbf{X}'\mathbf{D} \Big(\mathbf{X}\mathbf{a}_j + \mathbf{D}^{-1}\mathbf{y} - \mathbf{D}^{-1}(b'(\theta(\mathbf{a}, i)))_{:} \Big), && (2.37)
\end{aligned}$$

where we used

$$\begin{aligned}
\sum_i \nabla l\big(\theta(\mathbf{a}_j, i), \psi \big) &= \frac{1}{\psi} \sum_i \Big(y_i - b'(\theta(\mathbf{a}_j, i)) \Big) \nabla \theta(\mathbf{a}_j, i) \\
&= \frac{1}{\psi} \sum_i \Big(y_i - b'(\theta(\mathbf{a}_j, i)) \Big) (\mathbf{X}')_{:,i} \\
&= \frac{1}{\psi} \Big(\mathbf{X}'\mathbf{y} - \mathbf{X}'(b'(\theta(\mathbf{a}_j, i)))_{:} \Big),
\end{aligned}$$

it is not difficult to prove that

$$\boldsymbol{\beta} := \Big(\mathbf{X}'\mathbf{D}\mathbf{X} \Big)^{-1} \mathbf{X}'\mathbf{D} \Big(\mathbf{X}\mathbf{a}_j + \mathbf{D}^{-1}\mathbf{y} - \mathbf{D}^{-1}(b'(\theta(\mathbf{a}_j, i)))_{:} \Big)$$

is also the solution to the weighted least squares problem

$$\min_{\beta_k} \sum_i D_{ii} \big(z_i - \sum_k \beta_k h_k(\mathbf{x}_i) \big)^2,$$

with $z_i = \mathbf{X}_{i,:}\mathbf{a}_j + b''(\theta(\mathbf{a}_j, i))^{-1}\big(y_i - b'(\theta(\mathbf{a}_j, i))$.

As solving a weighted least squares problem can be done very efficiently, we can use the following algorithm to find the unique $\mathbf{a}$ solving Equation (2.31):

Algorithm 2 IRLS algorithm

(1) Initialize $\mathbf{a}_0$ - e.g randomly
(2) Until $\mathbf{a}_j$ does not change much, do:
Set

$$z_i := \mathbf{X}_{i,:}\mathbf{a}_j + b''(\theta(\mathbf{a}_j, i))^{-1}\big(y_i - b'(\theta(\mathbf{a}_j, i))\big)$$

Solve the weighted least squares problem

$$\min_{\beta_k} \sum_i D_{ii}\big(z_i - \sum_k \beta_k h_k(\mathbf{x}_i)\big)^2$$

with weights $D_{ii} := b''(\theta(\mathbf{a}_j, i))$
Set $\mathbf{a}_{j+1} = \boldsymbol{\beta}$

However in the case that $\mathbf{X}$ has less then full column rank the IRLS should still be the method of choice as can be seen by the following lemma which we will proof in Chapter 3.

Lemma 2.1. If $\mathbf{X}$ does not have full column rank then the Hessian is only negative semi-definite, meaning that under the assumption that at least one solution to (2.30) exists, uniqueness is not necessarily given anymore. If the IRLS algorithm converges to $\hat{\mathbf{a}}$ then it holds that we have $\mathbf{X}'\mathbf{y} = \mathbf{X}'(b'(\theta(\hat{\mathbf{a}}, i)))_:$, meaning that a solution to (2.31), namely $\hat{\mathbf{a}}$, has been found. The latter is also a global maximum of the objective function of (2.30), for any ψ.

Remark 14. There is another further natural extension of GLMs. It is called vector GLM (VGLM), and it allows for Y to be a multidimensional random vector belonging to a family with density depending on a multidimensional parameter $(\zeta_1, ..., \zeta_m)$. At the heart of GLMs was the mean function $g \equiv b'$ which established a link between the mean and the parameter θ. Basically it served the purpose that the image of $g(\sum_k a_k h_k(\cdot))$ exactly matched the range of the mean. This is the same for VGLMs, where one has for each parameter ζ_i now a function g_i, such that the image of $g_i(\sum_k a_{ki} h_{ki}(\cdot))$ exactly matches the range of ζ_i. Some more assumptions need to be made, especially about

the likelihood, in order to do something similar to the IRLS described above. More about this can be seen in [7].

2.2.3 Some inference results for GLMs

We will quickly discuss some model checking, some hypothesis testing and some model selection results. This subsection borrows heavily from the book by Wood [2]. Further results on inference for GLMs can be found for example in [8]. All the results in this subsection hold for the likelihood as well as the quasi-likelihood, see Chapter 3 - one only needs to replace l by q; also, these results hold for the canonical link case $g := b'$ as well as the more general case of using a different mean function g, see Chapter 3.

Residuals:

If we want to do model checking, having estimated ψ, there are two commonly used residual types. The first one are the Pearson residuals

$$\epsilon_i^p := \frac{y_i - \hat{\mu}_i}{\sqrt{V(\hat{\mu}_i)}}$$

and the second one are the deviance residuals

$$\epsilon_i^d := sign(y_i - \hat{\mu}_i)\sqrt{l(\eta(\mathbf{a}_{\max}, i)) - l(\eta(\hat{\mathbf{a}}, i))}$$

where $\hat{\mathbf{a}}_{\max}$ is defined below, $\hat{\mathbf{a}}$ is the estimated coefficient for the model $\eta(\mathbf{a}, i) := \sum_{k=1}^{m} a_k h_k(\mathbf{x}_i)$, see Chapter 3, $\hat{\mu}_i := g(\eta(\hat{\mathbf{a}}, i))$, $V(\hat{\mu}_i) := b''((b')^{-1}(\hat{\mu}))$ and $l(\eta(\mathbf{a}, i))$ is short for $l((b')^{-1}(g(\eta(\mathbf{a}, i)), \hat{\psi})$.
If the model is correct, then the Pearson residuals should be close to zero and have variance $\hat{\psi}$ - as $\sqrt{V(\hat{\mu}_i)} = \sqrt{\psi^{-1}Var(Y)}$. Usually one plots the fitted values $\hat{y}_i$ against the Pearson residuals and if there is a trend, then this is an indication for a misspecified model.
The deviance residuals are used in a similar manner. If the plot of the fitted values against ϵ_i^d is close to zero, has variance around one and the values experience no trend, then this could be an indication that the model is not misspecified.

Deviance:

An important quantity when doing tests on the parameters a_k is the so called deviance. Under given ψ, the deviance is defined by:

$$D(\mathbf{a}) := 2\psi \sum_i l(\eta(\mathbf{a}_{\max}, i)) - l(\eta(\mathbf{a}, i)),$$

where $\mathbf{a}_{\max}$ is the estimated parameter for the model that has as many coefficients as observations, thus n.

Remark 15. The motivation behind the deviance residuals is rather handwaving. For some special distributions belonging to the exponential family, in the large sample limit, we have $D(\hat{\mathbf{a}}) \sim \chi^2_{n-p}$. Heuristically speaking, as $D(\hat{\mathbf{a}})$ is a sum of $l(\eta(\mathbf{a}_{\max}, i)) - l(\eta(\hat{\mathbf{a}}, i))$ we could suspect that the signed square root of the latter is almost normally distributed.

Hypothesis test:

If we are interested in testing the hypothesis

$$H_0 : \mathbb{E}(Y) = g\big(\sum_k a_k h_k\big) \text{ vs } H_1 : \mathbb{E}(Y) = g\big(\sum_l a_l h_l\big),$$

where the first model is nested in the second one, then in the large sample limit, under H_0, we have

$$2\sum_i l(\eta(\hat{\mathbf{a}}_1, i)) - l(\eta(\hat{\mathbf{a}}_0, i)) \sim \chi^2_{p_1 - p_0},$$

where p_1 respectively p_0 are the traces of $\mathbf{X}_1(\mathbf{X}_1'\mathbf{D}_1\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{D}$ respectively $\mathbf{X}_0(\mathbf{X}_0'\mathbf{D}_1\mathbf{X}_0)^{-1}\mathbf{X}_0'\mathbf{D}$ and $\hat{\mathbf{a}}_1$ respectively $\hat{\mathbf{a}}_0$ are the estimated coefficients maximizing the likelihood for each model, with $\mathbf{D}_1$ respectively $\mathbf{D}_0$ being a diagonal matrix with elements $(\mathbf{D}_1)_{ii} = \frac{1}{V(\hat{\mu}_i)} g'(\eta(\hat{\mathbf{a}}_1, i))^2$ - $\hat{\mu}_i := g(\eta(\hat{\mathbf{a}}_1, i))$ - respectively $(\mathbf{D}_0)_{ii} = \frac{1}{V(\hat{\mu}_i)} g'(\eta(\hat{\mathbf{a}}_0, i))^2$ - $\hat{\mu}_i := g(\eta(\hat{\mathbf{a}}_0, i))$.
This result can for example be used to test if certain coefficients are zero.

Distribution of a_k:

Under the assumption that the model is true, it can be shown that in the large sample limit we get $\hat{\mathbf{a}} \sim \mathcal{N}(\mathbf{a}, (\mathbf{X}'\mathbf{D}\mathbf{X})^{-1}\psi)$, where $\mathbf{a}$ is the true underlying parameter.

Estimating ψ:

As mentioned before we need to know ψ to be able to use the deviance. Usually one estimates ψ from the data once a model has been fitted and

then goes from there.
In the large sample limit it holds that

$$\frac{1}{\psi}\sum_i \frac{(y_i-\hat{\mu}_i)^2}{V(\hat{\mu}_i)} \sim \chi^2_{n-p},$$

where $\hat{\mathbf{a}}$ are the estimated parameters for the model with p parameters maximizing the likelihood.

An estimate for ψ is thus based on

$$\frac{1}{n-p}\sum_i \frac{(y_i-\hat{\mu}_i)^2}{V(\hat{\mu}_i)}.$$

Model selection:

Model selection could be done by using either K-fold cross validation, which we will discuss in more detail for GAMs, or using the AIC. Using the AIC means that we look for the model that minimizes the following quantity:

$$-\sum_i l(\eta(\hat{\mathbf{a}}, i)) + p,$$

where p is the trace of $\mathbf{X}(\mathbf{X}'\mathbf{D}\mathbf{X})^{-1}\mathbf{X}'\mathbf{D}$ and $\hat{\mathbf{a}}$ are the estimated parameters maximizing the likelihood.

CHAPTER 3

Generalized additive models

3.1 GAMs and robust GAMs

In this section we will unify GLMs and AMs to what is known as generalized additive models (GAM). Furthermore, we will see how GAMs can be robustified, meaning that the estimated parameters will be less sensitive to outliers. This chapter borrows, yet again, heavily from [2].

Before we speak about how to unify the two approaches, let us widen our GLM framework we worked in so far, to the case when the canonical mean function b' is not defined on the whole real line.
An example would be the Gamma distribution, i.e. $b'(\theta) = -\frac{1}{\theta}$, where only negative θ are allowed; we see that even if we are able to get an estimate for this model, we would run into the problem of maybe having to restrict the space that $\mathbf{x}$ is defined on, so that $\sum_k \hat{a}_k h_k$ would never become zero or positive.

So, as a way out we could think of modelling the mean $\mathbb{E}(Y)$ differently, namely, we could try to find a function g which is defined on the whole real line $\mathbb{R}$ with the property that the image of g matches the range of $\mathbb{E}(Y)$; thus trying to model $\mathbb{E}(Y) = g(\sum_k a_k h_k)$. This means that we actually reparametrize the space of θ. The latter can be seen by looking at (2.27), which states

$$\mathbb{E}(Y) = b'(\theta).$$

We can thus deduce $g(\sum_k a_k h_k) = b'(\theta)$, and therefore this leads to the reparametrization $\theta = (b')^{-1}(g(\sum_k a_k h_k))$ - where we remind the reader that we assumed b' to be strictly monotone.

3.1.1 Penalized maximum likelihood

In Section 2.2 we considered the exponential family as an extension of the model we had looked at, namely we had assumed that $Y|X$ is normally distributed.
Let us assume again that we are in the GLM framework, that is, the density of $Y|X$ belongs to the exponential family, for fixed functions a, b and h, with parameters θ and ψ.
In view of subsection 2.2.2 and the observations made in the introduction of this chapter, we could therefore look at the problem

$$\max_{a_k,\psi} \sum_i l\Big(\theta\big(\sum_k a_k h_k(\mathbf{x}_i)\big), \psi\Big),$$

where we wrote $\theta(\sum_k a_k h_k(\mathbf{x}_i))$ for $(b')^{-1}(g(\sum_k a_k h_k))$ to keep the notation short.

As we had mentioned in Chapter 1, considering functions having the form $\sum_k a_k h_k(\mathbf{x}_i)$ - where $\mathbf{x}_i$ is high dimensional - can be very critical, as choosing $\boldsymbol{x}^{\boldsymbol{\alpha}}$ as basis functions leads to an explosion in the number of interactions and thus also parameters to estimate. In Section 2.1 we solved this problem, for the case of $Y|X$ being normally distributed, by considering additive models.

Going this way, the idea is thus to model θ as $\theta\big(f_1(X_1) + ... + f_p(X_p)\big)$. Therefore, we could consider the following problem

$$\max_{f_1,...,f_p,\psi} l\Big(\theta\big(f_1(X_1) + ... + f_p(X_p)\big), \psi\Big).$$

To estimate the functions $f_1, ..., f_p$, we could again take a Hilbert space approach, as it is done in [1], however it is much more convenient to instead take a penalized likelihood approach. In the spirit of AMs and splines we thus look at

$$\max_{f_k,\psi} \sum_{i=1}^{n} l\Big(\theta\big(f_1(x_{i1}) + ... + f_p(x_{ip})\big), \psi\Big) - \frac{1}{2\psi} \sum_{k=1}^{p} \lambda_k \int \left(\frac{\partial^2}{\partial x_k^2} f_k(x_k)\right)^2 dx_k, \tag{3.1}$$

for fixed $\lambda_k > 0$; the penalty forces the functions f_k to be smooth, as it was explained in Chapter 2. The reason for including ψ as well in the penalty, will become apparent as soon as we look at the Newton-Raphson procedure for solving this problem.

Again it is easy to see that any solution to problem (3.1) must be a sum of natural cubic splines. Assume that the functions $f_1, ..., f_p$ solve (3.1). Then if we choose for each f_k a natural cubic spline p_k, with nodes $x_{1k}, ..., x_{nk}$, which interpolates $(f_k(x_{1k}), x_{1k}), ..., (f_k(x_{nk}), x_{nk})$, we can replace

$$\sum_i l\Big(\theta\big(f_1(x_{i1}) + ... + f_p(x_{ip})\big), \psi\Big)$$

by

$$\sum_i l\Big(\theta\big(p_1(x_{i1}) + ... + p_p(x_{ip})\big), \psi\Big),$$

without changing the value of the objective function of (3.1). However, as already mentioned in the section about smoothing splines, a natural cubic spline, interpolating such points exactly, always has the property

$$\int \left(\frac{\partial^2}{\partial x_k^2} p_k(x_k)\right)^2 dx_k \leq \int \left(\frac{\partial^2}{\partial x_k^2} f_k(x_k)\right)^2 dx_k.$$

Therefore it follows that (3.1) is solved by natural cubic splines p_k with nodes in $x_{1k}, ..., x_{nk}$ - also see [2].

Remark 16. We would not necessarily need to model each function f_k as a smoothing spline. We could also consider interactions as well. However, we will start with this case for simplicity. How to treat the more general case is then considered in the subsection about identifiability.

So in the following it suffices to only look at $f_k = \sum_{j=1}^n a_{jk} h_{jk}$, where h_k are basis functions, such that f_k are natural cubic splines with nodes in $x_{1k}, ..., x_{nk}$.
In the following we will use again the definitions already introduced for the

AM case - but with radically different dimensions - namely

$$
\begin{aligned}
&\mathbf{X}^k := (h_{jk}(x_{ik}))_{i,j} && \in \mathbb{R}^{n\times n}\\
&\mathbf{X} := (\mathbf{X}^k_{i,:})_{i,k} = \begin{pmatrix} \mathbf{X}^1_{1,:} & \cdots & \mathbf{X}^p_{1,:} \\ \vdots & \ddots & \vdots \\ \mathbf{X}^1_{n,:} & \cdots & \mathbf{X}^p_{n,:} \end{pmatrix} && \in \mathbb{R}^{n\times n\cdot p}\\
&\mathbf{a}^k := (a_{1k}, ..., a_{nk})' && \in \mathbb{R}^{n\times 1}\\
&\mathbf{a} := (\mathbf{a}^1, ..., \mathbf{a}^p)' && \in \mathbb{R}^{n\cdot p\times 1}\\
&\mathbf{S}^k := \left(\int \left(\frac{\partial^2}{\partial x_k^2}(h_{jk}(x_k)h_{j'k}(x_k)) \right)^2 dx_k \right)_{j,j'} && \in \mathbb{R}^{n\times n}\\
&\mathbf{S} := \begin{pmatrix} \lambda_1\mathbf{S}^1 & 0 & \cdots \\ 0 & \ddots & 0 \\ \vdots & 0 & \lambda_p\mathbf{S}^p \end{pmatrix} && \in \mathbb{R}^{n\cdot p\times n\cdot p}.
\end{aligned}
$$

Furthermore, let us also introduce the following definitions which we will need just below:

$$
\begin{aligned}
&\mathbf{h} := (h_{11}, ..., h_{n1}, ..., h_{1p}, ..., h_{np})'\\
&\eta(\mathbf{a}, i) := \mathbf{a}'\mathbf{h}(\mathbf{x}_i)\\
&\mu_i := g(\eta(\mathbf{a}, i))\\
&V(\mu_i) := b''(\mu_i).
\end{aligned}
$$

We can rewrite (3.1) as

$$
\max_{a_{jk},\psi} \sum_i l\Big(\theta\big(f_1(x_{i1}) + ... + f_p(x_{ip})\big), \psi\Big) - \frac{1}{2\psi}\sum_k \lambda_k \int \left(\frac{\partial^2}{\partial x_k^2} f_k(x_k) \right)^2 dx_k \tag{3.2}
$$

$$
= \max_{\mathbf{a}^k,\psi} \sum_i l\Big(\theta\Big(\sum_k (\mathbf{a}^k)'\mathbf{X}^k_{i,:}\Big), \psi\Big) - \frac{1}{2\psi}\sum_k \lambda_k (\mathbf{a}^k)'\mathbf{S}^k\mathbf{a}^k \tag{3.3}
$$

$$
= \max_{\mathbf{a},\psi} \sum_i l\Big(\theta\big(\mathbf{a}'\mathbf{X}_{i,:}\big), \psi\Big) - \frac{1}{2\psi}\mathbf{a}'\mathbf{S}\mathbf{a} \tag{3.4}
$$

$$
= \max_{\mathbf{a},\psi} \sum_i \frac{y_i\theta_i - b(\theta_i)}{\psi} - h(y_i, \psi) - \frac{1}{2\psi}\mathbf{a}'\mathbf{S}\mathbf{a}, \tag{3.5}
$$

$$
\tag{3.6}
$$

where in the last line we only wrote θ_i instead of $\theta\big(\mathbf{a}'\mathbf{X}_{i,:}\big)$.

Differentiating the objective function in the latter, in a_{jk}, is basically similar to what we did for (2.30) to arrive at (2.31).
For each element of the sum we get

$$\begin{aligned}
\nabla(y_i\theta_i - b(\theta_i)) &= (y_i - b'(\theta_i))\nabla\theta_i \\
&= \Big(y_i - b'\Big((b')^{-1}(g(\eta(\mathbf{a},i)))\Big)\Big)\nabla(b')^{-1}(g(\eta(\mathbf{a},i))) \\
&= \Big(y_i - g(\eta(\mathbf{a},i))\Big)\frac{1}{b''((b')^{-1}(g(\eta(\mathbf{a},i))))}g'(\eta(\mathbf{a},i))\eta(\mathbf{a},i) \\
&= \Big(y_i - g(\eta(\mathbf{a},i))\Big)\frac{1}{b''(\mu_i)}g'(\eta(\mathbf{a},i))\nabla\eta(\mathbf{a},i) \\
&= \Big(y_i - g(\eta(\mathbf{a},i))\Big)\frac{1}{V(\mu_i)}g'(\eta(\mathbf{a},i))\nabla\eta(\mathbf{a},i).
\end{aligned}$$

Thus, if the term $\mathbf{a}'\mathbf{S}\mathbf{a}$ is also taken into account, which has the gradient $(\mathbf{S}' + \mathbf{S})\mathbf{a} = 2\mathbf{S}\mathbf{a}$ - because $\mathbf{S}$ is symmetric - we arrive at

$$\sum_i \Big(y_i - g(\eta(\mathbf{a},i))\Big)\frac{1}{V(\mu_i)}g'(\eta(\mathbf{a},i))\nabla_{\mathbf{a}}\eta(\mathbf{a},i) - \mathbf{S}\mathbf{a} = 0 \tag{3.7}$$

$$\frac{\partial}{\partial\psi}\sum_i l\big(\theta(\mathbf{a},i),\psi\big) = 0. \tag{3.8}$$

Again, Equation (3.7) is independent from ψ - this is actually the motivation for the ψ weight in front of the penalty; and this is justified by the fact that we will estimate the tuning parameters anyways. Therefore, just as we did with GLMs, we will again only concentrate on solving Equation (3.7) numerically - after which ψ will be estimated. This is done in the following subsection.

3.1.2 The P - IRLS algorithm

Before talking about which method to deploy for solving (3.7) let us quickly look at its solvability.

In the same way as it was done for GLMs, we could proof, by differentiating (3.7) once more, that the Hessian of

$$\sum_i l\Big(\theta\big(\mathbf{a}'\mathbf{X}_{i,:}\big),\psi\Big)$$

is given by

$$-\frac{1}{\psi}\mathbf{X}'\mathbf{D}\mathbf{X},$$

where $\mathbf{D}$ is the diagonal matrix made up of the elements

$$\mathbf{D}_{ii} := \left(\frac{g'(\eta(\mathbf{a}, i))^2}{V(\mu_i)} \alpha(i) \right) \tag{3.9}$$

with

$$\alpha(i) := 1 + (y_i - \mu_i) \left(\frac{V'(\mu_i)}{V(\mu_i)} - \frac{g''(\eta(\mathbf{a}, i))}{g'(\eta(\mathbf{a}, i))^2} \right);$$

see [3] for a proof.

Furthermore, it is easy to see that the Hessian of $\frac{1}{2}\mathbf{a}'\mathbf{S}\mathbf{a}$ is given by $\mathbf{S}$.
All in all, we thus have that the Hessian of the objective function is given by

$$-\frac{1}{\psi}\mathbf{X}'\mathbf{D}\mathbf{X} - \frac{1}{\psi}\mathbf{S}.$$

In the GAM case, contrary to GLMs, it is harder to tell what happens. First of all, the matrix $\mathbf{X}$ is now an $n \times np$ matrix, meaning that $\mathbf{X}$ has never full column rank. And secondly, another problem is that $\mathbf{D}$ can have negative entries - also they might depend on $\mathbf{a}$ now.
This means that it might not be so obvious if $\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S}$ is p.d, over the whole range of $\mathbf{a}$, or even only positive semi-definite; thus the objective function might not even be concave anymore - in this case one might rethink the modelling.
Also, it might help to keep the number of basis function fixed, so that the number of samples exceed the number of basis functions - if this makes sense should however be tested.
To check if $\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S}$ is p.d, in the case of $\mathbf{D}$ being constant, we could use the QR decomposition again.

Therefore we have the following. If the model assumptions are correct, meaning that there exists at least one solution to (3.7), then:

if: $\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S}$ is p.d everywhere $\rightarrow$ problem (3.7) has exactly one solution
if: $\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S}$ is not p.d everywhere $\rightarrow$ problem (3.7) might have multiple solutions which are not maxima.

Under the assumption that the model is correct and that $\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S}$ is p.d everywhere, we can find the unique maximum, in $\mathbf{a}$, of the objective function of (3.5) - which is the same for any ψ - approximately, again by resorting to

the Newton-Raphson method. This means that (3.7) is linearized; as we did already before in the case of GLMs. After some calculation this will give us:

$$\begin{aligned}
\mathbf{a}_{j+1} &= \mathbf{a}_j - \Big(\sum_i \mathcal{H}_l \left(\eta(\mathbf{a}_j, i), \psi\right) \Big)^{-1} \nabla_{\mathbf{a}} \Big(\sum_i l\big(\eta(\mathbf{a}_j, i), \psi\big) - \frac{1}{2\psi} \mathbf{a}_j' \mathbf{S} \mathbf{a}_j \Big) \\
&= \mathbf{a}_j + \left(\frac{1}{\psi} \mathbf{X}'\mathbf{D}\mathbf{X} + \frac{1}{\psi}\mathbf{S} \right)^{-1} \left(\frac{1}{\psi} \mathbf{X} \Big((y_i - \mu_i) \frac{1}{V(\mu_i)} g'(\eta(\mathbf{a}_j, i)) \Big)_{:} - \frac{1}{\psi} \mathbf{S}\mathbf{a}_j \right) \\
&= \left(\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S} \right)^{-1} \left(\mathbf{X}'\mathbf{D}\mathbf{X}\mathbf{a}_j + \mathbf{X} \Big((y_i - \mu_i) \frac{1}{V(\mu_i)} g'(\eta(\mathbf{a}_j, i)) \Big)_{:} \right) \\
&= \left(\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S} \right)^{-1} \mathbf{X}'\mathbf{D} \left(\mathbf{X}\mathbf{a}_j + \mathbf{D}^{-1} \Big((y_i - \mu_i) \frac{1}{V(\mu_i)} g'(\eta(\mathbf{a}_j, i)) \Big)_{:} \right)
\end{aligned}$$

where we used

$$\begin{aligned}
\psi \nabla_{\mathbf{a}} \sum_i l\big(\eta(\mathbf{a}_j, i), \psi\big) &= \sum_i \Big(y_i - \mu_i \Big) \frac{1}{V(\mu_i)} g'(\eta(\mathbf{a}_j, i)) \nabla_{\mathbf{a}} \eta(\mathbf{a}_j, i) \\
&= \mathbf{X} \Big((y_i - \mu_i) \frac{1}{V(\mu_i)} g'(\eta(\mathbf{a}_j, i)) \Big)_{:}.
\end{aligned}$$

Similar to GLMs it is easy to see that the latter is the solution to the weighted least squares problem with penalization:

$$\min_{\beta_{jk}} \sum_i D_{ii} \big(z_i - \sum_{jk} \beta_{jk} h_{jk}(\mathbf{x}_i) \big)^2 + \boldsymbol{\beta}'\mathbf{S}\boldsymbol{\beta},$$

where

$$\begin{aligned}
z_i &:= \mathbf{X}_{i,:}\mathbf{a}_j + \mathbf{D}^{-1}(y_i - \mu_i) \frac{1}{V(\mu_i)} g'(\eta(\mathbf{a}_j, i)) \\
&= \mathbf{X}_{i,:}\mathbf{a}_j + \frac{1}{g'(\eta(\mathbf{a}_j, i))^2} V(\mu_i) \alpha(i) (y_i - \mu_i) \frac{1}{V(\mu_i)} g'(\eta(\mathbf{a}_j, i)) \\
&= \mathbf{X}_{i,:}\mathbf{a}_j + \frac{1}{g'(\eta(\mathbf{a}_j, i))} \alpha(i) (y_i - \mu_i)
\end{aligned}$$

and $D_{ii} = \frac{g'(\eta(\mathbf{a}_j, i))^2}{V(\mu_i)} \alpha(i)$, as well as $\boldsymbol{\beta} := (\beta_{11}, ..., \beta_{n1}, ..., \beta_{1p}, ..., \beta_{np})'$.

This results in the following algorithm, called the P-IRLS, which approximately finds a solution to (3.7) - which may be a local maximum or local minimum - if the model is correct:

Algorithm 3 P-IRLS algorithm

(1) Initialize $\mathbf{a}_0$ - e.g randomly

(2) Until $\mathbf{a}_j$ or a quantity depending on $\mathbf{a}_j$ does not change much, do:
Set

$$z_i := \mathbf{X}_{i,:}\mathbf{a}_j + \frac{1}{g'(\eta(\mathbf{a}_j, i))}\alpha(i)(y_i - \mu_i)$$

Solve the weighted least squares problem

$$\min_{\beta_{jk}} \sum_i D_{ii}\big(z_i - \sum_k \beta_{jk} h_{jk}(x_{ik})\big)^2 + \boldsymbol{\beta}'\mathbf{S}\boldsymbol{\beta}$$

with weights $D_{ii} := \frac{g'(\eta(\mathbf{a}_j,i))^2}{V(\mu_i)}\alpha(i)$
Set $\mathbf{a}_{j+1} = \boldsymbol{\beta}$

Remark 17. An important thing to realize here is that the weights $\alpha(i)$ and thus $\mathbf{D}_{ii}$, as already mentioned before, might be negative, in which case we couldn't have derived the P-IRLS algorithm in this way.
One other possibility is that before inverting the Hessian we replace it with its expectation, namely the expectation of the Hessian of the log-liklihood - the Fisher information - which leads to the so called Fisher scoring algorithm. In this case we would get $\alpha(i) = 1$ instead for the P-IRLS algorithm - see [3].

Even if the matrix $\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S}$ is not always p.d we should use the P-IRLS algorithm anyways. First of all, no matrix inversion is necessary and, second of all, assuming that for each step we find a solution $\mathbf{a}_{j+1}$ - which is always the case for $\alpha(i) = 1$ - no matter if it is unique or not, we have:

$$\Big(\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S}\Big)\mathbf{a}_{j+1} = \mathbf{X}'\mathbf{D}\mathbf{X}\mathbf{a}_j + \mathbf{X}'\Big((y_i - \mu_i)\frac{1}{V(\mu_i)}g'(\eta(\mathbf{a}_j, i))\Big)_:$$

where $\mathbf{D}$ and $\mathbf{z}$ depend on $\mathbf{a}_j$. Assuming now that the P-IRLS algorithm converges, meaning $\mathbf{a}_j \to \hat{\mathbf{a}}$, we can see that the latter is equal to

$$\Big(\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S}\Big)\hat{\mathbf{a}} = \mathbf{X}'\mathbf{D}\mathbf{X}\hat{\mathbf{a}} + \mathbf{X}'\Big((y_i - \mu_i)\frac{1}{V(\mu_i)}g'(\eta(\hat{\mathbf{a}}, i))\Big)_:.$$

This however is exactly

$$\mathbf{X}'\Big((y_i - \mu_i)\frac{1}{V(\mu_i)}g'(\eta(\hat{\mathbf{a}}, i))\Big)_: - \mathbf{S}\hat{\mathbf{a}} = 0.$$

The last line is equal to (3.7), meaning that we have found a solution of the latter. We should then proceed to check if it is a maximum or a minimum - local or global. First we can look at the matrix $\mathbf{X}'\mathbf{D}(\hat{\mathbf{a}})\mathbf{X}+\mathbf{S}$, where we have explicitly written the dependence of $\mathbf{D}$ on $\hat{\mathbf{a}}$ now. If the latter is p.d then we have found a local maximum in $\mathbf{a}$, for any ψ, of the objective function of (3.5) - as the Hessian in this point is then negative definite. Further we can then check, which is for some members of the exponential family much easier than others, if the matrix $\mathbf{X}'\mathbf{D}(\mathbf{a})\mathbf{X}+\mathbf{S}$ is p.d over the whole range of $\mathbf{a}$. If this is so, we have found a global maximum, if it is only positive semi-definite, there may be more global maxima.

Theorem 3.1. *In case that the P-IRLS algorithm converges $\mathbf{a}_j \to \hat{\mathbf{a}}$ we have found a solution to (3.7), namely $\hat{\mathbf{a}}$. If $\mathbf{X}'\mathbf{D}(\hat{\mathbf{a}})\mathbf{X}+\mathbf{S}$ is p.d then this solution is a local maximum of the objective function, in $\mathbf{a}$, of (3.5) - for any ψ. Furthermore, if $\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S}$ is p.d for any $\mathbf{a}$, then we have found the unique maximum. If $\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S}$ is positive semi-definite for any $\mathbf{a}$, then we have found one global maximum.*

This means that the P-IRLS algorithm should be used as a method of choice, with possible multiple start overs to recover the maximal number of possible solutions.

Remark 18. In the P-IRLS algorithm we iterate until $\mathbf{a}_j$ or a quantity depending on $\mathbf{a}_{j+1}$ and $\mathbf{a}_j$ does not change much. Such a quantity could be for example the objective function of (3.1), or a measure depending on the estimated function values $\hat{f}_k^j$, by $\mathbf{a}_j$, namely

$$\Gamma(\mathbf{a}_{j+1}, \mathbf{a}_j) := \sum_i \psi l(\theta(a_j, i), \psi) - \psi h(y_i, \psi)$$

$$\Gamma(\mathbf{a}_{j+1}, \mathbf{a}_j) := \sum_k \frac{||\hat{f}_k^{j+1} - \hat{f}_k^j||}{||\hat{f}_k^j||}.$$

Remark 19. We could have chosen a slightly different approach to GAMs. As we have seen in the section about GLMs a solution to problem (2.30) fulfills Equation (2.31). We had looked at models of the form $\eta := \sum_k a_k h_k$ and then proceeded to the P-IRLS algorithm to get an estimate of $\mathbf{a}$. An approach to GAMs could now also be to start with a regular unpenalized maximum log-likelihood problem and then include the smoothing of the functions only at the very end in the IRLS algorithm. Although the two approaches start

with slightly different problems they both end up with the same estimation procedure - the P-IRLS. This way of approaching and solving GAMs, which is taken in [1], is called the local scoring algorithm. We will need this in the subsection about robust GAMs.

Remark 20. If b' is not defined on the whole of $\mathbb{R}$ and if we do not want to use another link g, we can still, in principle, use the P-IRLS. The Hessian would still be $-\frac{1}{\psi}\mathbf{X}'\mathbf{D}\mathbf{X} - \frac{1}{\psi}\mathbf{S}$, where now we always have a matrix $\mathbf{D}$ with positive diagonal elements, as $D_{ii} = b''(\eta(\mathbf{a}, i)) > 0$ - compare with GLMs. Thus the Hessian is at least negative semi-definite.
Assume that the parameter is only defined on an interval (c, d), where infinity is allowed, and that Equation (3.7) has a solution $\hat{\mathbf{a}}$. We can use the P-IRLS, for a starting value $\mathbf{a}_0$ which fulfills $\mathbf{a}_0'\mathbf{h}(\mathbf{x}_i) \in (c, d)$ for all i, to find a solution.
It follows that we have an optimization problem of a concave function over a convex set - where the latter is $\{\mathbf{a} \,|\, \mathbf{a}'\mathbf{x}_i \in (c, d)\, \forall i\}$ - because of the following. The objective function is at least concave in $\mathbf{a}$, maybe not strictly, because the Hessian is now at least negative semi-definite - by the above. Further, the set is convex because, for any two points $\tilde{\mathbf{a}}'\mathbf{x}_i$ and $\hat{\mathbf{a}}'\mathbf{x}_i$ which are in (c, d), for all i, we get that their line also must be in (c, d), for all i; as the latter is convex.
Thus a Newton-Raphson approach finds a global maximum as long as one exists - where it may be necessary to not take the full step $\mathbf{a}_{j+1} - \mathbf{a}_j$, so that we do not exit the feasible set.
One problem with this type of modelling is however that once the model is fitted, we might get some restrictions on the space of allowed $\mathbf{x}$; as $\hat{\mathbf{a}}'\mathbf{h}(\mathbf{x})$ must be in (c, d). This might lead to interpretability issues.

3.1.3 Identifiability

Suppose that we somehow can assume that it is more appropriate to model η by a slightly different model, namely

$$f(x_1, .., x_p) := \sum_{k_1} f_k(x_{k_1}) + \sum_{k_1 \neq k_2} f_{k_1 k_2}(x_{k_1}, x_{k_2}) + ... + \sum_{k_1 \neq ... \neq k_l} f_{k_1 ... k_l}(x_{k_1}, ..., x_{k_l}), \tag{3.10}$$

where all variables are continuous ones and some functions are zero; meaning that we would like to model interactions of up to l variables. Similar to before,

we start out from the following optimization problem

$$\max_{f_{k_1},...,f_{k_1...k_l},\psi} \sum_{i=1}^{n} l\Big(\theta\big(f(x_{i1},...,x_{ip})\big),\psi\Big) - \frac{1}{2\psi}\mathcal{R}(f_{k_1},...,f_{k_1...k_l}), \tag{3.11}$$

where $\mathcal{R}$ represents a penalty term - for smoothing some of the functions $f_{k_1},...,f_{k_1...k_l}$ - made up of penalties of the type (2.18). Assume now that this leads us to look for functions $f_{k_1},...,f_{k_1...k_l}$, having the form

$$\begin{aligned} f_{k_1} &= \sum_j a_{jk_1} h_{k_1} \\ &\vdots \\ f_{k_1...k_l} &= \sum_j a_{j_1 k_1...k_l} h_{k_1...k_l}, \end{aligned}$$

where some of the functions, maybe not all, are tensor product smoothing splines or smoothing splines - see [2] - corresponding to the penalty terms of $\mathcal{R}$; this is justified by the same arguments than for (3.1).

This means that plugging in the above into the penalty term, we end up with $\mathbf{a}'\mathbf{S}\mathbf{a}$, where $\mathbf{a}$ is the stacked vector of the parameters $a_{jk_1},...,a_{j_1 k_1...k_l}$. Finally, by defining

$$\eta(\mathbf{a},i) := \sum_{k_1}\sum_j a_{jk_1} h_{k_1}(x_{ik_1}) + ... + \sum_{k_1...k_l}\sum_j a_{j_1 k_1...k_l} h_{k_1...k_l}(x_{ik_1},...x_{ik_l}),$$

we can write problem (3.11) as

$$\max_{\mathbf{a},\psi} \sum_{i=1}^{n} \frac{y_i\theta_i - b(\theta_i)}{\psi} - h(y_i,\psi) - \frac{1}{2\psi}\mathbf{a}'\mathbf{S}\mathbf{a}, \tag{3.12}$$

with a positive semi-definite matrix $\mathbf{S}$; where one should keep in mind that $\theta_i = (b')^{-1}(g(\eta(\mathbf{a},i)))$.

Remark 21. The step from problem (3.11) to problem (3.12) is only there to somewhat justify the form of problem (3.12) and the form of the functions $f_{k_1},...,f_{k_1...k_l}$. Of course we could have already started out with the latter, as our analysis will not be effected by (3.11); we only need the penalized likelihood problem (3.12) in the following. The same of course holds for anything we have done so far in the GAM framework.

From here on out, we can do exactly the same analysis as above for problem (3.1) - as we only needed a penalty term translating to $\mathbf{a}'\mathbf{S}\mathbf{a}$, where $\mathbf{S}$ is positive semi-definite. It therefore follows that we can also employ the P-IRLS algorithm without problem. If the P-IRLS algorithm converges, we have found a solution, under certain circumstances - see Theorem 3.1.

However, for the case discussed so far, there are multiple solutions, because we have multiple representations for the functions $f_{k_1}, ..., f_{k_1 \dots k_l}$.
This is very undesirable. Having multiple representations for $f_{k_1}, ..., f_{k_1 \dots k_l}$ decreases the interpretability of our model - e.g. if we have two different functions $\tilde{f}_1$ and $\bar{f}_1$ for the same effect x_1, this does not allow us to pinpoint the exact effect that x_1 has on f.
We can see, by the same arguments as in Chapter 2, that adding a constant to any of the functions and subtracting it from another one, does not change the function f. Additionally, as we consider interactions, a further problem is that by adding, for example, a function $x_1 \mapsto h(x_1)$ to f_1 and subtracting it from f_{12} does yet again not change f.
Thus, for problem (3.12), we already know from the beginning on, that - in the case that there is a solution - there are multiple solutions. This also entails that there is no use in checking if $\mathbf{X}'\mathbf{D}(\mathbf{a})\mathbf{X} + \mathbf{S}$ is p.d everywhere - compare with Theorem (3.1).
Of course this ambiguity is a consequence of our modelling.
The way we model f, see (3.10), leaves too much freedom. We would like to put single effects of variables into the functions f_k and second order interactions - and only such, meaning without single effects - into $f_{k_1 k_2}$; and then higher order ones into $f_{k_1 \dots k_l}$, etc.
It is therefore necessary to make this model identifiable and remove these issues. Thus, we would have to put, on the one hand, constraints on the parameters corresponding to constant basis functions h_{jk_1} or $h_{jk_1 \dots k_l}$ and, on the other hand, put constraints on the parameters corresponding to basis functions corresponding to single effects, double interaction effects, etc.
Also, we might want to consider some conditional constraints on the parameters, for example, we may only want to work with cyclic spline, meaning that start and endpoint of the spline is the same; this can be useful for time stamped data.
Such constraints can always be given in the form of $\mathbf{F}\mathbf{a} = 0$, where $\mathbf{F}$ is a $\mathbb{R}^{m \times p}$ matrix - with p being the dimension of $\mathbf{a}$, and $m < p$.

As it is explained in [2], looking at the QR decomposition of $\mathbf{F}'$

$$\mathbf{F}' = \mathbf{Q}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix},$$

where $\mathbf{Q} \in \mathbb{R}^{p\times p}$ is an orthogonal matrix and $\mathbf{R} \in \mathbb{R}^{m\times m}$ is an upper triangular matrix, we have that a vector $\mathbf{a}$ is a solution of $\mathbf{Fa} = 0$ iff $\mathbf{a} = \mathbf{Q}_{:,(m+1):p}\,\mathbf{b}$ for a unique $\mathbf{b} \in \mathbb{R}^{p-m}$.
Thus we should substitute $\mathbf{a}$ by $\mathbf{Q}_{:,(m+1):p}\,\mathbf{b}$ in problem (3.12) and then proceed with P-IRLS.

We summarize all this in the following

Lemma 3.1. Assume that we want to solve problem (3.12) and that $\mathbf{F}$ is a matrix putting linear constraints on the parameters. Furthermore, assume that the P-IRLS algorithm, for the following problem, converges:

$$\max_{\mathbf{b},\psi} \sum_i \frac{y_i\theta(\mathbf{Q}_{:,(m+1):p}\,\mathbf{b}, i) - b(\theta(\mathbf{Q}_{:,(m+1):p}\,\mathbf{b}, i))}{\psi} - h(y_i, \psi) - \frac{1}{2\psi}\mathbf{b}'\tilde{\mathbf{S}}\mathbf{b}, \tag{3.13}$$

with $\tilde{\mathbf{S}} := \mathbf{Q}'_{:,(m+1):p}\mathbf{S}\mathbf{Q}_{:,(m+1):p}$ and call the solution $\hat{\mathbf{b}}$. Then we have that $\hat{\mathbf{a}} = \mathbf{Q}_{:,(m+1):p}\,\hat{\mathbf{b}}$ solves the corresponding maximum likelihood equation - compare with (3.7) - and we can use Theorem 3.1, for $\hat{\mathbf{a}}$, to determine if the latter is a global or local maximum - or in the worst case a minimum - and if it is unique; for any ψ.

Remark 22. It makes more sense to remove identifiability issues beforehand, instead of doing so with a matrix $\mathbf{F}$. This means that we should set some parameters already to zero, once we have set up all the basis functions, to remove these issues. However, the lemma above also allows us to include different kinds of constraints to remove identifiability issues and more.

Remark 23. It is important to realize that, in the case that we have removed all identifiability issues, we can not deduce that $\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S}$ is positive definite - not even in the case where $\mathbf{D}$ is constant.
It is easy to see this. For example, for the model $f(x_1, x_2) = a_{11} + a_{21}x_1 + a_{31}x_1^2 + a_{22}x_2 + a_{32}x_2^2$, we get, in the case that we have collinearity between x_1 and x_2 - meaning that there exists a constant a such that $x_1 = ax_2$ - $f(x_1, x_2) = a_{11} + a_{21}ax_2 + a_{31}a^2x_2^2 + a_{22}x_2 + a_{32}x_2^2$. Thus, the second column

of the model matrix $\mathbf{X}$ is always proportional to the fourth column and, likewise, the third is proportional to the fifth. This means that the matrix $\mathbf{X}$ does not have full column rank - so $\mathbf{X}'\mathbf{D}\mathbf{X}$ is not p.d.

To end this section, let us quickly talk about how to integrate factor variables into the GAM framework as well. For example, look at the model

$$f(x_1,..,x_p) := \sum_{k_1} f_k(x_{k_1}) + \sum_{k_1 \neq k_2} f_{k_1 k_2}(x_{k_1}, x_{k_2}) + ... + \sum_{k_1 \neq ... \neq k_l} f_{k_1 ... k_l}(x_{k_1}, ..., x_{k_l}), \tag{3.14}$$

where some variables might be factor variables now. Without loss of generality, assume that x_1 is a factor variable with r levels $1, ..., r$. This means, that in the above model, we would have

$$f_1(x_1) = \begin{cases} c_1 \text{ if } x_1 = 1 \\ \vdots \\ c_r \text{ if } x_1 = r \end{cases},$$

where $c_1, ..., c_r$ would be parameters to estimate.
Likewise, without loss of generality, a function $f_{12}(x_1, x_2)$, where x_2 is a continuous variable, would lead to

$$f_{12}(x_1, x_2) = \begin{cases} f^1(x_2) \text{ if } x_1 = 1 \\ \vdots \\ f^r(x_2) \text{ if } x_1 = r \end{cases}.$$

For all factor variables and their levels, we can then impose, for each factor and each level, on each function containing a continuous variable a penalization term, e.g. (2.18). For the example above we could penalize each f^i, for $i = 1, .., r$, so that these will be smoothing splines. Everything we have said so far then still holds. However, identifiability becomes an even bigger issue now - as it is known from simple linear regression with factors.
Also, one needs to keep in mind that the parameters to estimate basically grow times r for each function where a factor variable, with r levels, is included - for functions with two factor variables, with levels $1, ..., r_1$ and $1, ..., r_2$, this would be $r_1 r_2$; and so on.

3.1.4 Degrees of freedom, smoothing parameters, confidence intervals and the quasi-likelihood

It is very important to find good tuning parameters $\lambda_1, ..., \lambda_k$, because if they are chosen too big or too small this can result in too much or too lit-

tle smoothing. A first approach is taken by establishing a link between the tuning parameters and the degrees of freedom. After this we will see a more founded way of estimating the tuning parameters. Mainly this will be done by cross validation. At last, we will also address the question of confidence intervals and see a nice way of how to extend our methodology to responses Y not belonging to the exponential family.

In what follows, we will almost always motivate the quantities and definitions we look at by the Gaussian, constant variance, AM case, and then only state their generalizations to the GAM case, without any proofs or such. For a more thorough treatment we refer to [1], [2] and [3].

Degrees of freedom:

To begin with, let us motivate the notion of degrees of freedom for GAMs.

Let us go back to the least squares problem (2.3), where Y_i was independently normally distributed, with constant variance.
Furthermore, let

$$\mathbf{X} = \mathbf{U} \begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{V}' \tag{3.15}$$

be the singular value decomposition of $\mathbf{X}$ - where the latter might not have full column rank - meaning that $\mathbf{U} \in \mathbb{R}^{n \times n}$ respectively $\mathbf{V} \in \mathbb{R}^{p \times p}$ are orthogonal matrices and $\mathbf{\Sigma} = \text{diag}(\sigma_1, ..., \sigma_r) \in \mathbb{R}^{r \times r}$ is a diagonal matrix, with diagonal elements $\sigma_i > 0$, being the singular values of the matrix $\mathbf{X}$, for $i = 1, ..., r$ - ordered from highest to lowest.

Assume now that we know that the true underlying function is of the form $f = \sum_i a_k h_k$. Then we could, by restricting ourselves to the points $\mathbf{x}_1, ..., \mathbf{x}_n$, interpret f also as function from $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ to $\mathbb{R}$. We could thus identify f with $(f(\mathbf{x}_1), ..., f(\mathbf{x}_n))'$ and do the following, by using (3.15):

$$\begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{pmatrix} = \begin{pmatrix} h_1(\mathbf{x}_1) & \cdots & h_p(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_n) & \cdots & h_p(\mathbf{x}_n) \end{pmatrix} \mathbf{a} = \mathbf{U} \begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{V}'\mathbf{a} = \sum_{i=1}^{r} c_i \mathbf{u}_i,$$

where $\mathbf{c}' = (c_1, ..., c_r)' := \mathbf{a}'(\sum_{i=1}^{r} \sigma_i \mathbf{v}_i)$ and $\mathbf{u}_i$ respectively $\mathbf{v}_i$ is the i-th column of $\mathbf{U}$ respectively $\mathbf{V}$. Now the vectors $\mathbf{u}_i$ can also be interpreted as functions from $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ to $\mathbb{R}$. In this sense, this means that we represent

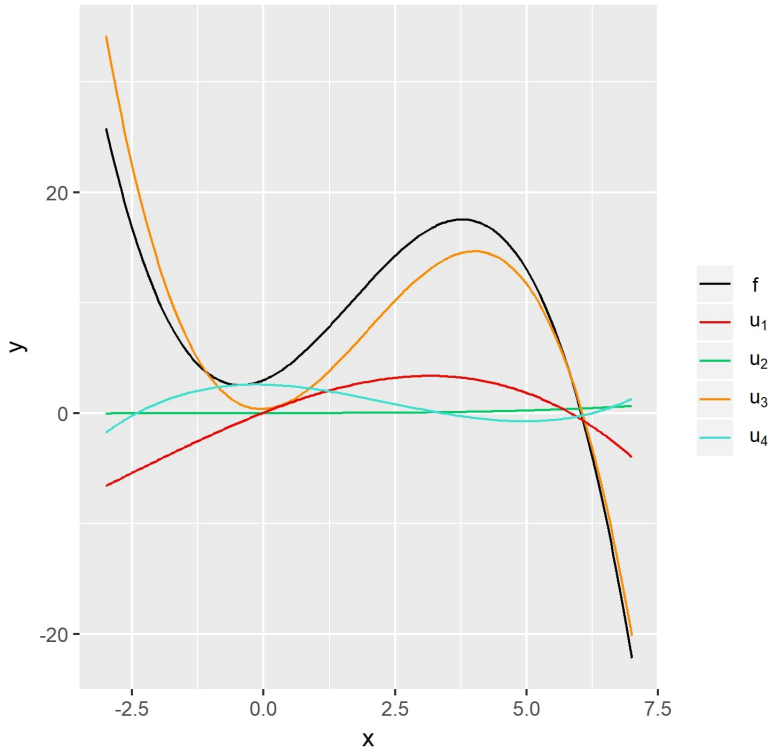


Figure 3.1: Decomposition of $f(x) = 3 + 2x + 2x^2 - 0.4x^4$

f, an n-dimensional vector, as a linear combination of p basis vectors, therefore f is actually only p dimensional - see Figure 3.1 for an example.

Interestingly, in the case that $\mathbf{X}$ has full column rank, $r = p$, we get that $\text{tr}(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') = p$ holds.
Thus if our model is correct, $\text{tr}(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')$ gives us the correct number of basis functions or degrees of freedom; also indicating that $\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ leaves the number of basis functions unchanged.

Remark 24. It is also important to notice that least squares sets $n - p$ basis functions to zero, as $(f(\mathbf{x}_1), ..., f(\mathbf{x}_n))'$ is only p dimensional. This must not be the case for GAMs - see below.

In analogy to the above one can define the degrees of freedom for GAMs in a similar fashion, namely as $df(\lambda_1, ..., \lambda_k) := \text{tr}(\mathbf{X}(\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}'\mathbf{D})$, provided that $\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S}$ is invertible. Hopefully, $df(\lambda_1, ..., \lambda_k)$ is a good measure

of the true underlying dimension.

If we use smoothing splines in the GAM framework, we cannot, of course, choose the number of basis functions ourself, only the tuning parameters $\lambda_1, ..., \lambda_k$; however we suspect that making the latter very big will reduce the possible number of basis functions as we then opt for very smooth straight functions. Therefore, heuristically speaking, if we want to force the model to have a certain dimension p, or if we wanted to fix it, we could try to approximately solve $df(\lambda_1, ..., \lambda_k) = p$ for $\lambda_1, ..., \lambda_k$.

There are many motivations for the definition above, not only is there the analogy to the usual linear model, but also the following observation can be taken into account.
For the GAM, we had upon convergence $\hat{\mathbf{a}} = (\mathbf{X'DX} + \mathbf{S})^{-1}\mathbf{X'Dz}$. Using the singular value decomposition of $\mathbf{D}^{\frac{1}{2}}\mathbf{X}$, and for simplicity, w.l.o.g, assuming that $\mathbf{X} \in \mathbb{R}^{n \times m}$, with $m > n$, has rank n, we get:

$$
\begin{aligned}
&\mathbf{X}(\mathbf{X'DX} + \mathbf{S})^{-1}\mathbf{X'D} \\
&= \mathbf{D}^{-\frac{1}{2}}\mathbf{U}\begin{pmatrix}\boldsymbol{\Sigma} & \mathbf{0}\end{pmatrix}\mathbf{V'}\left(\mathbf{V}\begin{pmatrix}\boldsymbol{\Sigma} \\ \mathbf{0}\end{pmatrix}\mathbf{U'U}\begin{pmatrix}\boldsymbol{\Sigma} & \mathbf{0}\end{pmatrix}\mathbf{V'} + \mathbf{S}\right)^{-1}\mathbf{V}\begin{pmatrix}\boldsymbol{\Sigma} \\ \mathbf{0}\end{pmatrix}\mathbf{U'D}^{-\frac{1}{2}} \\
&= \mathbf{D}^{-\frac{1}{2}}\mathbf{U}\begin{pmatrix}\boldsymbol{\Sigma} & \mathbf{0}\end{pmatrix}\mathbf{V'}\left(\mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V'} + \mathbf{S}\right)^{-1}\mathbf{V}\begin{pmatrix}\boldsymbol{\Sigma} \\ \mathbf{0}\end{pmatrix}\mathbf{U'D}^{-\frac{1}{2}} \\
&= \mathbf{D}^{-\frac{1}{2}}\mathbf{U}\begin{pmatrix}\boldsymbol{\Sigma} & \mathbf{0}\end{pmatrix}\mathbf{V'V}\boldsymbol{\Sigma}^{-1}\left(\mathbf{I} + \boldsymbol{\Sigma}^{-1}\mathbf{V'SV}\boldsymbol{\Sigma}^{-1}\right)^{-1}\boldsymbol{\Sigma}^{-1}\mathbf{V'V}\begin{pmatrix}\boldsymbol{\Sigma} \\ \mathbf{0}\end{pmatrix}\mathbf{U'D}^{-\frac{1}{2}} \\
&= \tilde{\mathbf{U}}\begin{pmatrix}\mathbf{I}_n & \mathbf{0}\end{pmatrix}\left(\mathbf{I} + \boldsymbol{\Sigma}^{-1}\mathbf{V'SV}\boldsymbol{\Sigma}^{-1}\right)^{-1}\begin{pmatrix}\mathbf{I}_n \\ \mathbf{0}\end{pmatrix}\tilde{\mathbf{U}}' \\
&= \tilde{\mathbf{U}}_{:,1:n}\left(\mathbf{I} + \boldsymbol{\Sigma}^{-1}\mathbf{V'SV}\boldsymbol{\Sigma}^{-1}\right)^{-1}\tilde{\mathbf{U}}'_{:,1:n},
\end{aligned}
$$

where $\tilde{\mathbf{U}} := \mathbf{D}^{-\frac{1}{2}}\mathbf{U}$. Furthermore, as $\boldsymbol{\Sigma}^{-1}\mathbf{V'SV}\boldsymbol{\Sigma}^{-1}$ is symmetric and positive semi-definite we can find an orthogonal matrix $\mathbf{Q}$ s.t $\boldsymbol{\Sigma}^{-1}\mathbf{V'SV}\boldsymbol{\Sigma}^{-1} = \mathbf{Q'ZQ}$, where $\mathbf{Z}$ is a diagonal matrix with $z_i > 0$, ordered from highest to lowest - z_i are the eigenvalues of $\boldsymbol{\Sigma}^{-1}\mathbf{V'SV}\boldsymbol{\Sigma}^{-1}$, thus depending on $\lambda_1, ..., \lambda_k$ - on the diagonal.

So all in all we can write

$$\mathbf{X}(\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S})^{-1}\mathbf{X}'\mathbf{D}\mathbf{z} = \tilde{\mathbf{U}}_{:,1:n}\left(\mathbf{I}+\boldsymbol{\Sigma}^{-1}\mathbf{V}'\mathbf{S}\mathbf{V}\boldsymbol{\Sigma}^{-1}\right)^{-1}\tilde{\mathbf{U}}'_{:,1:n}\mathbf{z}$$
$$= \sum_{i=1}^{n}\tilde{\mathbf{u}}_i\frac{1}{1+z_i}\tilde{\mathbf{u}}_i'\mathbf{z}.$$

This means that the fitted values are a sum of n basis functions which are shrunk - as $z_i > 0$; this is slightly different from before where some of the coefficients where entirely set to zero. Also, we can see that our suspicion from before is confirmed - as in the case that $\lambda_k \to +\infty$, for $\mathbf{S}_k$ being p.d, we get $z_k \to +\infty$, and so, for big λ_k some of the terms in the above sum will be close to zero.
Here we could thus argue that the trace, which is the sum of the eigenvalues, $\sum_i \frac{1}{1+z_i}$, gives us an indication of the dimension of the fitted function; strictly speaking, the dimension is still n but the basis elements are shrunk therefore being diminished in its "dimension".

Further arguments which support the definition above can be for example found in [2] or [4].

Smoothing parameters:

We will now turn to the question of how $\lambda_1, ..., \lambda_k$ could be chosen or estimated by a different approach; as for the one above we need to approximately solve large linear systems, namely $df(\lambda_1, ..., \lambda_k) = p$, which can be very costly.

As we will need it soon, let us just quickly mention that usually one estimates ψ, upon P-IRLS convergence - for which ψ does not come into play - by

$$\hat{\psi} = \frac{1}{n-p}\sum_i V(\hat{\mu}_i)^{-1}(y_i - \hat{\mu}_i)^2,$$

where p is $\mathrm{tr}((\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S})^{-1}\mathbf{X}'\mathbf{D}\mathbf{X})$ and $\hat{\mu}_i := g(\eta(\hat{\mathbf{a}}, i))$ - see [2] or [3].

The Gaussian AM case:

To begin with we look at a criterion which assesses the goodness of a model - for fixed $\lambda_1, ..., \lambda_k$ - which would then provide us with the means of comparing different models for different tuning parameters.

Let the setting be the Gaussian AM one again, with canonical mean function - the identity.
A first measure to be able to judge the performance of a model could be to consider the so called expected mean squared error (MSE), namely

$$\mathbb{E}\left(||\,\mathbb{E}(\mathbf{y}) - \mathbf{X}\hat{\mathbf{a}}||^2\right),$$

which measures how close the model mean fit comes to the true underlying mean - on the given observations; meaning that $\mathbf{X}$ is not independent on $\hat{\mathbf{a}}$.

In the Gaussian case it is not hard to show that, see [2],

$$\frac{1}{n}\,\mathbb{E}\left(||\,\mathbb{E}(\mathbf{y}) - \mathbf{X}\hat{\mathbf{a}}||^2\right) = \frac{1}{n}\,\mathbb{E}\left(||\mathbf{y} - \mathbf{P}\mathbf{y}||^2\right) - \sigma^2 + \frac{2}{n}\mathrm{tr}(\mathbf{P})\sigma^2 \tag{3.16}$$

holds; where $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}'$, see (2.21), is the so called hat matrix with $\mathbf{P}\mathbf{y} = \mathbf{X}\hat{\mathbf{a}}$.

So

$$\frac{1}{n}||\mathbf{y} - \mathbf{P}\mathbf{y}||^2 - \sigma^2 + \frac{2}{n}\mathrm{tr}(\mathbf{P})\sigma^2$$

could be used as an estimation for the MSE. However, as it is pointed out in [2], this is only a good estimator if σ^2 is known. In the case that σ^2 is indeed known, we could thus look for tuning parameters $\lambda_1, ..., \lambda_k$ which minimize the quantity above.

Remark 25. If one does not know σ^2 and it is estimated by $\hat{\sigma}^2 = \frac{1}{n-\mathrm{tr}(\mathbf{P})}||\mathbf{y} - \mathbf{P}\mathbf{y}||^2$, then (3.16) could at least still be used as a pointer for a good model. Rearranging (3.16) a bit, and assuming that for a good model $\frac{1}{\sigma^2}\,\mathbb{E}\left(||\mathbf{y} - \mathbf{P}\mathbf{y}||^2\right) \approx \frac{1}{\hat{\sigma}^2}||\mathbf{y} - \mathbf{P}\mathbf{y}||^2$ should hold, we get

$$\begin{aligned}\frac{1}{\sigma^2}\,\mathbb{E}\left(||\,\mathbb{E}(\mathbf{y}) - \mathbf{X}\hat{\mathbf{a}}||^2\right) &= \frac{1}{\sigma^2}\,\mathbb{E}\left(||\mathbf{y} - \mathbf{P}\mathbf{y}||^2\right) - n + 2\mathrm{tr}(\mathbf{P}) \\ &\approx \frac{1}{\hat{\sigma}^2}||\mathbf{y} - \mathbf{P}\mathbf{y}||^2 - n + 2p \\ &= n - p - n + 2p = p.\end{aligned}$$

This suggests that we should look for models with

$$\frac{1}{\hat{\sigma}^2}||\mathbf{y} - \mathbf{P}\mathbf{y}||^2 - n + 2p \approx p,$$

where the left side is the so called Mallow's C_p.

As mentioned before, if σ^2 is not known we should not use the MSE, on the observations used for the fit, to asses goodness of fit.

Assume that we have some data - e.g $(\mathbf{y}_j, \mathbf{x}_j)$, for $j = 1, ..., n$ - left, which we have not used for estimation. Then one could of course consider approximating with these observations the quantity $\mathbb{E}(||E(\mathbf{Y}) - \mathbf{X}'\hat{\mathbf{a}}||^2)$ - for independent data $(\mathbf{Y}, \mathbf{X})$ of the ones we used to fit. However, it makes more sense to look at a very similar measure: the so called mean squared prediction error (PSE)

$$\mathbb{E}\left(||\mathbf{Y} - \mathbf{X}'\hat{\mathbf{a}}||^2\right),$$

again for independent data of the ones used to fit, as it is quite simple to estimate the PSE by $\sum_j (y_j - \mathbf{x}_j'\hat{\mathbf{a}})^2$.

Nevertheless, one problem remains, namely in some situations we cannot afford to not use observations for estimation. Luckily there is a way out of this. The PSE is approximated by what is known as cross validation.
The idea is to use all but one datum $(y_i, \mathbf{x}_i)$ for the estimation of $\mathbf{a}$, and then use this remaining datum - thus constituting a new datum - to get an estimate of the PSE. This can be done for any i and thus we could consider

$$\frac{1}{n}\sum_i (y_i - \mathbf{x}_i'\hat{\mathbf{a}}_i^{-1})^2, \tag{3.17}$$

where $\hat{\mathbf{a}}_i^{-1}$ are the estimated parameters of $\mathbf{a}$ using all but the i-th datum $(y_i, \mathbf{x}_i)$.
This can be however quite expensive, as we need to perform n fits. Luckily, it can be shown that the latter is equal to the so called OCV score,

$$\frac{1}{n}\sum_i \frac{(y_i - \mathbf{x}_i'\hat{\mathbf{a}})^2}{(1 - \mathbf{P}_{ii})^2},$$

where $\mathbf{P}_{ii}$ is the i-th diagonal element of the projection matrix $\mathbf{P}$ and $\hat{\mathbf{a}}$ are the estimated parameters when using all the data. For a proof of this see [2].

Unfortunately, as it is pointed out in [2], this quantity has the drawback of not being invariant to transformations under orthogonal matrices $\mathbf{Q}$. Meaning that transforming the data $\mathbf{y}$ and $\mathbf{X}$ by $\mathbf{Q}$ leads to the same estimated parameters $\hat{\mathbf{a}}$, which can easily be seen by looking at (2.21), but different OCV scores. This is however critical, as we can always transform our data and thus obtain different OCV scores for the same problem.

Therefore one modifies the OCV score, by replacing $\mathbf{P}_{ii}$ with $\mathrm{tr}(\mathbf{P})n^{-1}$, to make it rotation invariant. This leads to the so called GCV score, namely

$$\frac{n||\mathbf{y}-\mathbf{X}\hat{\mathbf{a}}||^2}{(n-\mathrm{tr}(\mathbf{P}))^2}, \tag{3.18}$$

which is easily seen to be rotational invariant as matrices commute under the trace; again see [2] for a more extensive discussion on this.
All in all, we could then choose $\lambda_1, ..., \lambda_k$ such that (3.18) is minimized.

Remark 26. As it is mentioned in [3], OCV and GCV have the tendency to be not sensitive enough when it comes to overfit. It can be argued that using

$$\frac{n||\mathbf{y}-\mathbf{X}\hat{\mathbf{a}}||^2}{(n-\zeta\mathrm{tr}(\mathbf{P}))^2},$$

for a fixed $\zeta > 0$, rather than (3.18) could help to prevent this.

Remark 27. There is also the possibility of leaving out more than one datum. Mainly, we could split the data into K almost equally big sets and perform fitting on $K-1$ of them and testing on the remaining one. Such an approach is generally known as K-fold cross validation and can help to reduce computational cost - for the approach described above we would have $K = n$. However one should keep in mind that using a bigger K can increase bias but decrease variance - and vice versa, meaning that smaller K has lower bias but bigger variance; see [4].

The general GAM case:

For the general case we could just as well derive the GCV score (3.18) from

$$\frac{1}{n}\sum_i (y_i - \hat{\mu}_i^{-1})^2,$$

with $\hat{\mu}_i^{-1} = g(\eta(\hat{\mathbf{a}}_i^{-1}, i))$ - $\eta(\mathbf{a}, i) := \mathbf{a}'\mathbf{h}(\mathbf{x}_i)$ - where $\hat{\mathbf{a}}_i^{-1}$ is the estimate obtained by P-IRLS by leaving out the datum $(y_i, \mathbf{x}_i)$. We would then equally arrive at

$$\frac{n||\mathbf{y}-\hat{\boldsymbol{\mu}}||^2}{(n-\mathrm{tr}(\mathbf{P}))^2},$$

where now $\mathbf{P} := \mathbf{X}(\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S})^{-1}\mathbf{X}'\mathbf{D}$, for $\mathbf{D}$ as described in the P-IRLS algorithm for $\hat{\mathbf{a}}$ and $\hat{\mu}_i = g(\eta(\hat{\mathbf{a}}, i))$.

However, as it is mentioned in [3], one should keep in mind that this score tends to under-smooth in some cases, e.g. binary data.

There are many other scores which can help us estimate $\lambda_1, ..., \lambda_k$, which also tend to deliver better results.
For example, as the deviance - see subsection (2.2.3) - namely $D(\mathbf{a}) = -\sum_i l(\eta(\mathbf{a}, i))\psi + \text{const}$, is a "natural" way to measure distance, in the case of GAMs, to the "best" model, we could try to look for tuning parameters which maximize

$$\frac{1}{n}\sum_i l(\eta(\hat{\mathbf{a}}_i^{-1}, i)),$$

remembering that $l(\eta(\mathbf{a}, i))$ is short for $l((b')^{-1}(g(\eta(\mathbf{a}, i))), \psi)$; basically we approximate the maximum value of the expectation of the log-likelihood $\mathbb{E}(l(\eta(\mathbf{a})))$ here.

Going this way one can argue, as it is done from OCV to GCV, and arrive at

$$n\frac{D(\hat{\mathbf{a}})}{(n-p)^2},$$

for unknown ψ - compare this with (3.18); we remark that $D(\hat{\mathbf{a}})$ is independent of ψ by definition.

For known ψ we can do something similar to the MSE and get as measure

$$D(\hat{\mathbf{a}}) + 2\psi p.$$

For more information about these two measures we refer to [2] and [1]. Wood, see [3], also suggests another one, namely

$$\frac{D(\hat{\mathbf{a}})}{n} + \frac{2}{n}\frac{p}{n-p}\sum_i \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}.$$

For computational and numerical considerations of these measures we also refer to [3].

Confidence intervals:

In the following we will assume that the number of basis functions m does not change with the number of observations n; so that all the large sample results hold.

Let us now turn to the distribution of $\mathbf{a}$ in the big sample limit, and consequently to confidence intervals.

We know that in the Gaussian AM case, with constant variance, that the unique solution to (2.20) is given by

$$\hat{\mathbf{a}} = (\mathbf{X}'\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}'\mathbf{y},$$

provided that $\mathbf{X}'\mathbf{X} + \mathbf{S}$ is p.d.

This means that $\hat{\mathbf{a}}$ is also Gaussian, with mean and variance:

$$\begin{aligned}\mathbb{E}(\hat{\mathbf{a}}) &= (\mathbf{X}'\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}'\mathbf{X}\mathbf{a} \\ \mathbb{V}\mathrm{ar}(\hat{\mathbf{a}}) &= \sigma^2(\mathbf{X}'\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X} + \mathbf{S})^{-1};\end{aligned}$$

where we see that in the case with zero penalty, we have $\hat{\mathbf{a}} \sim \mathcal{N}(\mathbf{a}, (\mathbf{X}'\mathbf{X})^{-1}\sigma^2)$.

Thus, in the case that we have a penalty unequal to zero, we usually get $\mathbb{E}(\hat{\mathbf{a}}) \neq \mathbf{a}$. This however means that it makes not much sense to use the latter to construct confidence intervals or make inference - as the results would be bias; especially for $\lambda_k \to +\infty$ we get $\mathbb{E}(\hat{\mathbf{a}}) \to \mathbf{0}$ - this also shows that a penalty introduces different degrees of bias.

Another approach would be to take a Bayesian point of view, as it is described in [2]. We could try to consider the improper prior, namely

$$f(\mathbf{a}) \propto e^{-\frac{1}{2}\mathbf{a}'\mathbf{S}\mathbf{a}}.$$

Under this prior, and by looking at the quantity $\mathbf{z} := \mathbf{X}\hat{\mathbf{a}} + \mathbf{D}^{-1}\Big((y_i - \mu_i)\frac{1}{V(\mu_i)}g'(\eta(\hat{\mathbf{a}}, i))\Big)_i$ given by a converged P-IRLS - this quantity is especially attractive as it is calculated in each iteration - it is proven in [2] that, under mild conditions, we have in the large sample limit, under given ψ,

$$\mathbf{a} \sim \mathcal{N}\big(\hat{\mathbf{a}}, (\mathbf{X}'\mathbf{D}\mathbf{X} + \mathbf{S})^{-1}\psi\big),$$

where $\mathbf{D}$ is the diagonal matrix obtained in the converged P-IRLS algorithm, namely $\mathbf{D}_{ii} := \frac{g'(\eta(\hat{\mathbf{a}},i))^2}{V(\hat{\mu}_i)}\alpha(i)$.

With this result at hand it is now easy to construct confidence intervals, for example for $\hat{f}(\mathbf{x}) := \hat{\mathbf{a}}'\mathbf{h}(\mathbf{x})$. Also we can obtain, by sampling, quantities depending on $\mathbf{a}$.

Hypothesis test:

Again we assume that the dimension of basis functions is fixed.

As we have seen above, in the Gaussian case, it can be deduced

$$\hat{\mathbf{a}} \sim \mathcal{N}((\mathbf{X}'\mathbf{X}+\mathbf{S})^{-1}\mathbf{X}'\mathbf{X}\mathbf{a}, \sigma^2(\mathbf{X}'\mathbf{X}+\mathbf{S})^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X}+\mathbf{S})^{-1}).$$

It is noted in [2], that in the general GAM case, in the large sample limit, we have:

$$\hat{\mathbf{a}} \sim \mathcal{N}(\mathbb{E}(\hat{\mathbf{a}}), \psi(\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S})^{-1}\mathbf{X}'\mathbf{D}\mathbf{X}(\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S})^{-1}).$$

Furthermore, if we would like to test

$$H_0 : \tilde{\mathbf{a}} = \mathbf{0} \text{ vs } \tilde{\mathbf{a}} \neq \mathbf{0}$$

for a subvector $\tilde{\mathbf{a}}$ of $\mathbf{a}$, we can use that, in the large sample limit, it holds:

$$\hat{\tilde{\mathbf{a}}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}),$$

for an appropriate submatrix $\mathbf{\Sigma}$ of $\psi(\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S})^{-1}\mathbf{X}'\mathbf{D}\mathbf{X}(\mathbf{X}'\mathbf{D}\mathbf{X}+\mathbf{S})^{-1}$ - see [2].

The latter result can then be used to construct a chi-square test or such.

The quasi-likelihood:

We quickly want to mention the quasi-likelihood approach.

Sometimes, when doing modelling, one can not be sure if the response variable Y truly does belong to an exponential family, as we have always required so far, however one still has an idea about how the mean and variance behave.
Provided that we have a mean-variance relationship, meaning that we know that for each datum y_i, its mean $\mu_i := \mathbb{E}(Y_i|X_i)$ and variance $\mathbb{V}\text{ar}(Y_i|X_i)$ exist, and that the latter two satisfy the relation

$$\mathbb{V}\text{ar}(Y_i|X_i) = V(\mu_i)\psi,$$

for some function V and a constant $\psi > 0$ - that is to say that there is a link between the mean and the variance - then one can define the log-quasi likelihood of y_i as

$$q_i(\mu_i) := \int_{y_i}^{\mu_i} \frac{y_i - x}{\psi V(x)} dx,$$

provided that this integral always exists for any possible μ_i.
Oddly enough the sum

$$\sum_i q_i(\mu_i)$$

possesses many of the same properties of the log-likelihood. The most important one being that when modelling $\mu_i = g(\sum_k a_k h_k)$, for an appropriate function g, the gradient of $\sum_i q_i(g(\eta(\mathbf{a}, i)))$, with $\eta(\mathbf{a}, i) := \sum_k a_k h_k(\mathbf{x}_i)$, is equal to

$$\frac{1}{\psi} \sum_i \frac{(y_i - \mu_i)}{V(\mu_i)} g'(\eta(\mathbf{a}, i)) \nabla_{\mathbf{a}} \eta(\mathbf{a}, i).$$

Setting this to zero is the same as (3.7), without the penalty term - but which can be included in the definition of the sum of quasi-log likelihood without problem.

This means that we can use the P-IRLS algorithm of Chapter 3 again to find its solutions and therefore estimates of $\mathbf{a}$; as only (3.7) was used for its derivation.
In the big sample limit the quasi-likelihood behaves similar to the log-likelihood. For further discussions and exact results we refer to [3].

Remark 28. It is interesting to observe that if we have a random variable Y, coming from a family of probability distributions, only depending on the mean and the variance, with the property that its shift and scale is again in the same family, then we can reconsider $\mathbb{E}\left(\left(\frac{Y-\mu(X)}{V(X)}\right)^2\right)$ - for the case in which the mean respectively the variance is a function $\mu(X)$ respectively $V(X)$. In such a case $\frac{\mathbf{Y}-\mu(X)}{V(X)}$ would be iid and therefore we could plug in the sample distribution, leading to

$$\frac{1}{n} \sum_i \left(\frac{y_i - \mu(\mathbf{x}_i)}{V(\mathbf{x}_i)} \right)^2;$$

compare this with the remark at the end of the introduction of Chapter 2.
Differentiating this in μ and setting it to zero, basically also leads to (3.7) - without penalty.

3.2 Robust GAMs

3.2.1 A robust GAM version for response outliers

In this subsection we will discuss a robust version of GAMs. Mainly we follow the paper by A. Alimadad and M. Salibian-Barrera, see [5], but first let us consider the following.

In many applications it can occur that the data contains outliers - e.g a mistake in the measurement, a wrong comma when data is collected and recorded or just an occurrence of a very unlikely event. Heuristically said, an outlier in the response variable Y is a very unlikely observation - in the normal case for example, outliers are points which are very far away from the mean - or an observation which deviates from the expected behaviour of most of the other points.
It is however important to not only recognize outliers, as they can represent important events which occur, but also to robustify the methods we have talked about so far, so that the estimates do not change too much in the presence of such. This must happen automatically, as for higher dimensional cases it is impossible to look at a plot to recognize outliers and remove the latter by hand - also this would be very subjective.

To begin with, assume that we are interested to model the mean of a Gaussian variable Y, with constant variance, by an AM approach; as it was described in Chapter 2.
More precisely, we would look for parameters $\mathbf{a}$ solving the following problem

$$\min_{\mathbf{a}} \|\mathbf{y} - \mathbf{X}\mathbf{a}\|^2 + \mathbf{a}'\mathbf{S}\mathbf{a} = \min_{\mathbf{a}} \sum_i (y_i - \mathbf{h}(\mathbf{x}_i)'\mathbf{a})^2 + \mathbf{a}'\mathbf{S}\mathbf{a}.$$

Changing just one observation now, for example y_1, can have a severe effect on the estimated parameters $\mathbf{a}$. This can be seen by looking at its solution $\hat{\mathbf{a}} = (\mathbf{X}'\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}'\mathbf{y}$ - (2.21) - which can also be written as a linear combination of vectors $\mathbf{v}_i$ -which are the columns of $(\mathbf{X}'\mathbf{X}+\mathbf{S})^{-1}\mathbf{X}'$ - thus $\sum_i \mathbf{v}_i y_i$. Provided that $\mathbf{v}_1$ is not zero, we see that $\hat{\mathbf{a}}$ can become arbitrarily big by letting $y_1 \to +\infty$.

The main reason why this can happen is that the objective function of the above minimization problem contains the Euclidean distance. Heuristically speaking, if y_1 becomes very big in comparison to the other points, then $\mathbf{h}(\mathbf{x}_1)'\mathbf{a}$ needs to adapt enough to account for this misbehaviour - therefore distorting $\hat{\mathbf{a}}$.

This can lead to catastrophic effects, which can be readily seen in the very simple linear case with only one outlier.
So an idea could be to replace the Euclidean distance with another measure of distance which is more robust. Intuitively, very big quantities should be damped, or given less weight, so that in the presence of an outlier y_1, the quantity $\mathbf{h}(\mathbf{x}_1)'\mathbf{a}$ does not have to adjust too much to keep the objective function as minimal as it is without the outlier. This means that we could look at

$$\min_{\mathbf{a}} \sum_i \rho(y_i - \mathbf{h}(\mathbf{x}_i)'\mathbf{a}) + \mathbf{a}'\mathbf{S}\mathbf{a},$$

where ρ is an appropriate loss function measuring distance.

A very important loss function is the so called Huber loss function:

$$\rho(u) := \begin{cases} \frac{1}{2}u^2 & |u| \leq c \\ c(|u| - \frac{1}{2}c) & |u| > c \end{cases}$$

with its derivative being - we will need this later -

$$\psi(u) := \begin{cases} u & |u| \leq c \\ c \text{ sign}(u) & |u| > c \end{cases},$$

where c is an appropriately chosen constant, see [5]; Figure 3.2 shows ρ, ψ_c and $\frac{1}{2}u^2$ for $c = 0.5$.

We will now turn back to [5]. In the latter Alimadad and Salibian-Barrera consider a robustified version of the GAM approach to detect outliers considering disease outbreaks.
It is demonstrated that their approach seems to work very well for the Poisson family and the binomial distribution.
Only outliers in the response are considered, however this is the important case, as GAMs seem to be very sensitive to outliers in the response.

For the Gaussian case, with constant variance, we can just replace in the above minimization problem the Euclidean distance with ρ. In the case of GAMs, an important realization is that we could have also arrived at Equation (3.7), if we would have started out with the function

$$\frac{1}{\psi}\sum_{i=1}^{n}\left(\frac{y_i - \mu_i}{\sqrt{V(\mu_i)}}\right)^2 + \frac{1}{\psi}\mathbf{a}'\mathbf{S}\mathbf{a}, \tag{3.19}$$

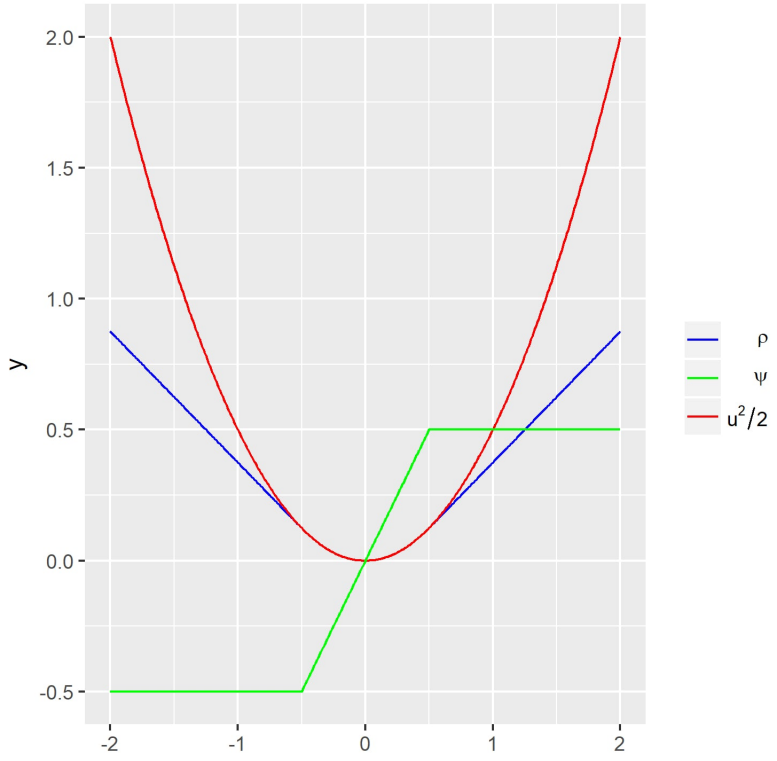


Figure 3.2: Huber loss function ρ, its derivative ψ and $\frac{1}{2}u^2$

then would have taken its derivative in μ and would have set it to zero - assuming that $V(\mu_i)$ is constant. Actually, the IRLS and P-IRLS are motivated this way in [2].

So we could replace the square function in (3.19) by ρ as suggested by our considerations above and this would basically lead us to the equation Alimadad and Salibian-Barrera start out from, namely

$$\sum_{i=1}^{n} \psi_c\left(\frac{y_i - \mu_i}{\sqrt{V(\mu_i)}}\right) \frac{1}{\sqrt{V(\mu_i)}} \nabla_{\mathbf{a}} \mu_i - b_n(\mathbf{a}) = 0, \tag{3.20}$$

where $b_n(\mathbf{a}) = \sum_i \mathbb{E}\left(\psi_c\left(\frac{y_i-\mu_i}{\sqrt{V(\mu_i)}}\right)\frac{1}{\sqrt{V(\mu_i)}}\nabla_{\mathbf{a}}\mu_i\right)$ - except for b_n, which is just a correction factor for unbiasedness, and penalty $\mathbf{S}$, which is zero here but which can be included in the algorithm later on - see Remark 19 .

Remark 29. Actually, Alimadad and Salibian-Barrera mention that their motivation for starting from Equation (3.20) comes from a quasi-likelihood approach, rather than maximum log-likelihood one - however this amounts to the same Equation (3.20).

The authors then proceed to using a Fisher-scoring approach - which is just Newton-Raphson where one replaces the Hessian with the Information matrix - to solve (3.20).
This leads to an iterative approach where at each step $\mathbf{a}_{j+1}$ must fulfill

$$\mathbf{X}'\mathbf{D}\mathbf{X}\mathbf{a}_{j+1} = \mathbf{X}'\mathbf{D}\mathbf{z}_i \tag{3.21}$$

where $\mathbf{D}$ is a diagonal matrix with elements $\mathbf{D}_{ii}$ - $\mathbf{D}_{ii}$ and $\mathbf{z}_i$ are described below.

Comparing this with the unpenalized GLM/GAM case - in the GAM case with $\mathbf{S} = \mathbf{0}$ - one can see that this is the same than what we did for solving (3.7); except that the weights $\mathbf{D}_{ii}$ and the vector $\mathbf{z}_i$ here are different now.

With this at hand it is easy to argue that we can just translate (3.21) into a weighted and penalized least squares problem again.

We will now state the robust generalized local scoring algorithm as derived by the authors and refer for all proofs to their paper [5].

Algorithm 4 Robust Generalized Local Scoring Algorithm

(1) Initialize $t = 0$, $f_0^0 = g^{-1}(\tilde{y})$, $f_k^0 = 0$, with $\tilde{y} = median_{1 \leq i \leq n} y_i$, $\eta_i^0 = f_0^0$ and $\mu_i^0 = \tilde{y}$

(2) Increment $t \leftarrow t + 1$
Calculate the following quantities for $i = 1, ..., n$:

$$r_i^t = \frac{y_i - \mu_i^t}{\sqrt{V(\mu_i^t)}}$$
$$h_i^t = \psi_c(r_i^t) - \mathbb{E}(\psi_c(r_i^t))$$
$$l_i^t = \left(\frac{\mathbb{E}(\psi_c'(r_i^t))}{\sqrt{V(\mu_i^t)}} g'(\eta_i^t) + \frac{1}{2} \mathbb{E}(\psi_c'(r_i^t) r_i^t) \frac{1}{V(\mu_i^t)} V'(\mu_i^t) g'(\mu_i^t) + \mathbb{E}\left(\frac{\partial}{\partial \eta_i} \mathbb{E}(\psi_c(r_i^t)) \right) \right)$$
$$\mathbf{D}_{ii}^t = l_i^t \frac{1}{\sqrt{V(\mu_i^t)}} g'(\eta_i^t)$$
$$z_i^t = \eta_i^t + \frac{h_i^t}{l_i^t}$$

(3) Now solve the following weighted penalized least squares problem

$$\min_{\beta_{jk}} \sum_{i=1}^{n} \mathbf{D}_{ii}^t (z_i^t - \sum_{k=1}^{p} \beta_{jk} h_{jk}(x_{ik}))^2 + \boldsymbol{\beta}' \mathbf{S} \boldsymbol{\beta},$$

and set $f_k^{t+1} = \sum_j \beta_{jk} h_{jk}$ for $k = 1, .., p$.

(5) Calculate a convergence criterion $\Gamma(f_1^{t+1}, ..., f_p^{t+1}, f_1^t, ..., f_p^t)$.

(6) If the criterion is smaller than a fixed $\epsilon > 0$ stop, otherwise calculate $\eta_i^{t+1} := f_0^0 + \sum_k f_k(x_{ik})$ and $\mu_i^{t+1} := g(\eta_i^{t+1})$ and go to (2).

Remark 30. The latter algorithm is implemented in the R package "rgam".

To end this subsection we want to mention at last that the estimation of the tuning parameters $\lambda_1, ..., \lambda_k$ is not so trivial in the case of outliers. As it is mentioned in [5], even when taking the robust estimation described above, the estimation of the tuning parameters may still be sensitive to outliers - for some of the measures discussed at the end of the last chapter.

They thus suggest two different measures to avoid this, namely

$$\sum_i \psi_c\left(\frac{y_i - \hat{\mu}_i^{-1}}{\sqrt{V(\hat{\mu}_i^{-1})}}\right)^2 \text{ and } \sum_i w\big(d_i\big)$$

where $w : \mathbb{R} \to \mathbb{R}$ is a bounded non-negative function and $d_i = 2\big(l(\eta(\mathbf{a}_{\max}), i) - l(\eta(\hat{\mathbf{a}}_i^{-1}, i))\big)$.

Remark 31. Alternatively to the approach described in this subsection, one could also try to damper the influence of outliers directly in the maximum likelihood problem (3.1) by composing the likelihood, or the deviance, with an appropriate function ρ as described before. Such an approach is taken by Hastie and Tibshirani in [1].

3.3 An application of GAMs

In the following we will look at three examples. For the first two we will look at simulated data to illustrate GAMs and see what can go wrong when the model is misspecified. For the third example we will consider real data coming from the UCI Machine Learning Repository.

3.3.1 A simulated Gaussian example

Without outliers

Let us look at first at the following model:

$$Y(x, z) = \sin(8x) - 2z^3 + 0.2z^2 + z + \epsilon, \tag{3.22}$$

where $\epsilon \sim \mathcal{N}(0, 0.1)$. We simulate $n = 100$ uniformly distributed independent points x_i and z_i, between 0 and 1, which we then use to construct a grid on $[0, 1]^2$. With the latter gridpoints we further simulate ϵ independently and then construct $Y(x, z)$ with (3.22).

Using the implemented gam function of the mgcv and the ggplot2 package we obtain the following resulting plots:

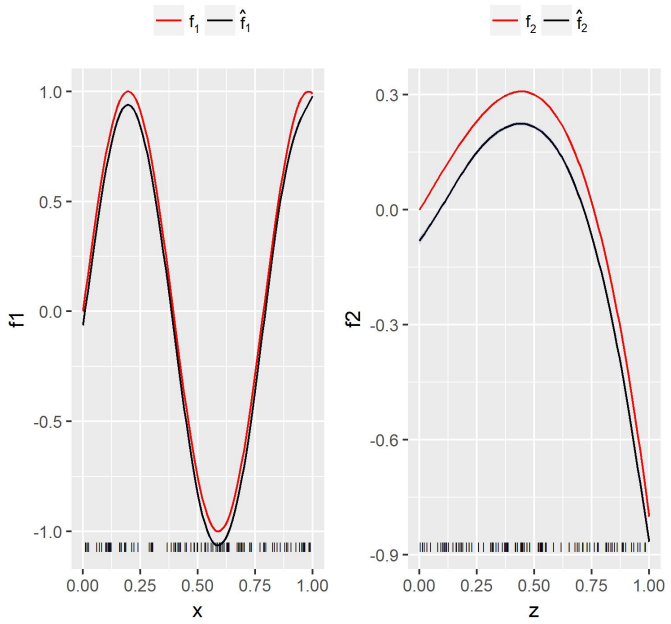
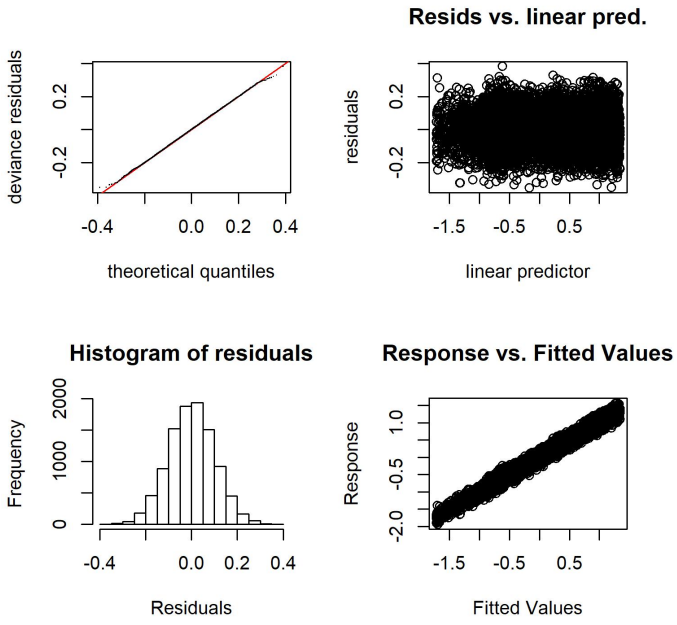


Figure 3.3: Plot of estimate $\hat{f}_1$ respectively $\hat{f}_2$ of $f_1(x) := \sin(8x)$ respectively $f_2(z) := -2z^3 + 0.2z^2 + z$. The ticks at the bottom represent the simulated x_i and z_i.



We can see that the estimated functions $\hat{f}_1$ and $\hat{f}_2$ come quite close to the true functions f_1 and f_2 - except for the addition of a constant for f_2, which is accounted by the fact that by using the gam function of the mgcv we also get an intercept term. We note as well that $f_1(x) = sin(8x)$ is very well approximated by a natural cubic spline $\hat{f}_1$.
Also the deviance residual plots clearly show that the latter are very close to normal and that the linear predictor plotted vs. the residuals experience no trend. This is a strong hint to the fact the model is not misspecified.

With outliers

Now let us quickly see what happens if we add outliers to the above artificial example. More specifically, we will replace 100 of the total $n^2 = 10^4$ points with outliers - so in total $\frac{10^2}{10^4} = \frac{1}{10}$-th of the data is corrupted. We construct these outliers by simulating $Y(x, z)$ with a noise $\epsilon \sim \mathcal{N}(0, 500)$, for 10^2 randomly picked points, instead of $\epsilon \sim \mathcal{N}(0.01)$.

Fitting a non-robust GAM to this we get the following plot:

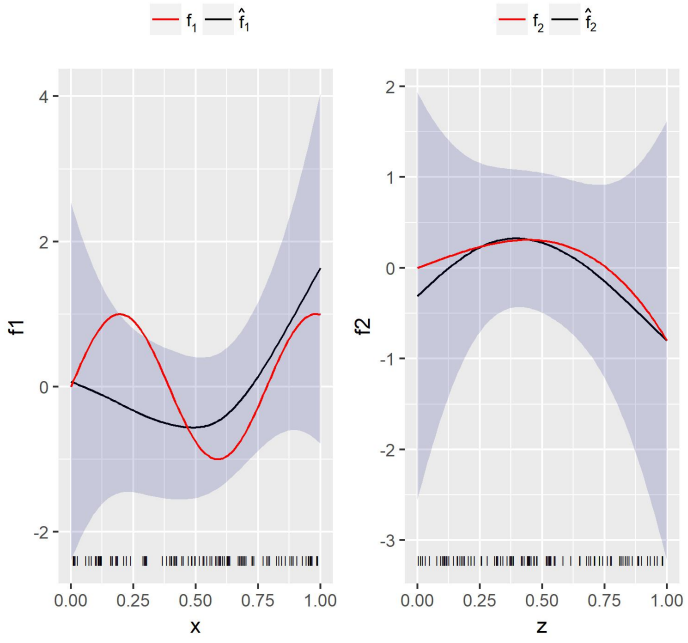


Figure 3.4: Non robust estimation of f_1 and f_2

We can see that the estimated function $\hat{f}_1$ is now far away from the truth. Also the confidence intervals "explode" in comparison to the uncorrupted case. This means that a robustified GAM method is absolutely necessary in this case.

So let us try to fit a robustified version of GAMs. As the rgam package does not support, so far, the case of Y being Gaussian, we use the RBF package which implements the methods discussed in [6], by Boente, Martinez and Salibian–Barrera. We will not discuss the methods they explored in their paper here and only mention that they do not take an approach based on splines but rather build upon the Hilbert space approach described in Chapter 2 to arrive at a robust version for additive models resulting in a robust backfitting algorithm.

So, using the corresponding robust GAM method implemented in the RBF package we get the following resulting plot:

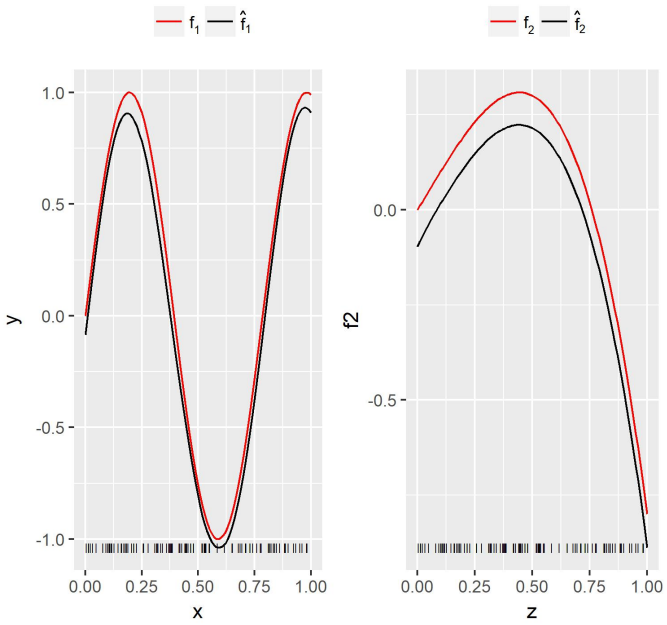


Figure 3.5: Robust estimation of f_1 and f_2

We can see that the estimate of $\hat{f}_1$ is much better now and not far away from what we had gotten in the no outlier case, compare Figure 3.5 with Figure 3.3. There is only a small discrepancy left between f_1 respectively f_2 and $\hat{f}_1$

respectively $\hat{f}_2$, mainly due to not taking the intercept into account.

3.3.2 A simulated binomial example

Without outliers

As a second example we simulate, as for the first example, $n = 100$ uniformly distributed independent points x_i and z_i, between 0 and 1, which we then use to construct a grid on $[0, 1]^2$.
However, this time we look at a binomially distributed, more specifically a Bernoulli distributed, variable $Y(x, z)$ satisfying the following relationship:

$$\mathbb{E}(Y|X, Z)(x, z) = \cos(8x) - 2z^3 + 0.2z^2 + z + 0.5xz^2;$$

thus with $f_1(x) = cos(8x)$ and $f_2(z) = -2z^3 + 0.2z^2 + z$. This means that for each grid point (x, z) we simulate 1 with probability $\frac{\exp(f_1(x)+f_2(z)+0.5xz^2)}{1+\exp(f_1(x)+f_2(z)+0.5xz^2)}$.

Using again the mgcv and the ggplot2 package we obtain the following plots.

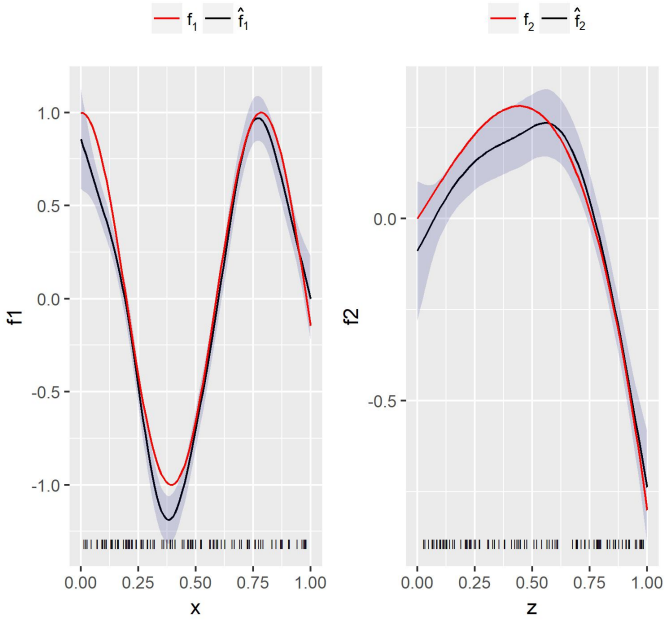
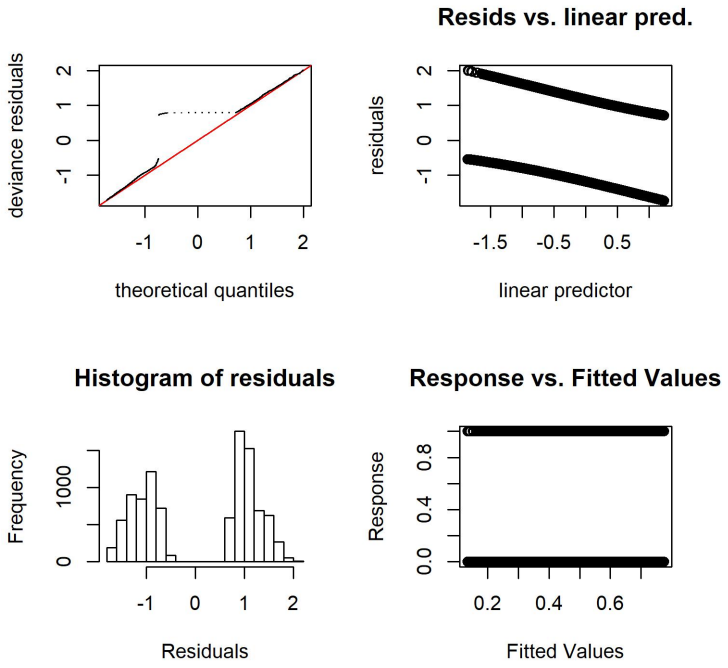
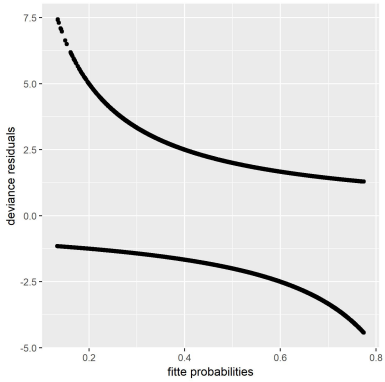


Figure 3.6: Plot of estimate $\hat{f}_1$ respectively $\hat{f}_2$ of $f_1(x) := \cos(8x)$ respectively $f_2(z) := -2z^3 + 0.2z^2 + z$ and the corresponding confidence bands. The ticks at the bottom represent the simulated x_i and z_i.



We can see from Figure 3.6 that even though the model is misspecified, as we have only modelled f_1 and f_2 and not the interactions, the estimates for f_1 respectively f_2 are quite close to the truth and capture the trends very well.
Furthermore, by looking at the residual plots, we can clearly see that the deviance residuals are non-normal. This is however no concern in the case of a binomial distribution for GAMs. As it is also mentioned in [2] it is hard to do model checking with these plots. In the binomial case the model can be specified correctly while these same plots, with deviance residuals not being normal etc, would indicate a misspecification in the Gaussian case.
However we can see from the frequency-residual plot that there seem to be more residuals accumulating around 1 than around -1. This, as the binomial distribution is symmetric, could be a hint that the model is slightly misspecified. Also notice that, in the case of binomial data, the fact that the linear predictor vs residuals plot clearly shows a trend, is normal. Furthermore, looking at the fitted probabilities vs residuals plot we might be led to think that there is a problem at high and low probabilities.



At last, we could also look at the misclassification rate, which is 0.35, meaning that we generate $m = 10$ more data and then, by constructing a classifier, where any prediction $\hat{f}$ over 0.5 gets assigned 1 and the rest gets assigned 0, we can compare it with the true probabilities $P(Y = 0)$ and $P(Y = 1)$ and calculate the rate of how often we'd misclassify. This rate is quite high and could now lead us to think that our model is very wrong. However it is important to keep in mind that, in this example, many predicted probabilities are around 0.5 and so it might not be that easy. We could also look at another measure, namely $\frac{1}{m}\sum((f(x_i, z_i) - \hat{f}(x_i, z_i))^2)$, which is always between 0 and 1, where the closer to zero the better. In this case we'd obtain 0.23. All in all, we could say that the obtained model is an adequate fit, however it might not be ideal for using it as a classifier - as explained above.

With outliers

Again let us look at what happens if we add outliers to this dataset. More specifically, we replace the points (x, z) with x lying between 0.45 and 0.5 - where the true function f_1 has a local minimum, see Figure 3.6 - with outliers. This is done by adding a noise, namely a uniformly distributed $\epsilon \sim \mathcal{U}[0, 10]$, to f_1 before simulation of 0 and 1; thus now with a success probability of $\frac{\exp(f_1(x)+\epsilon+f_2(z)+0.5xz^2)}{1+\exp(f_1(x)+\epsilon+f_2(z)+0.5xz^2)}$, for each grid point (x, z). In total we get that our dataset contains 8% outliers.

We obtain the following solution plots using the usual non-robust GAM approach:

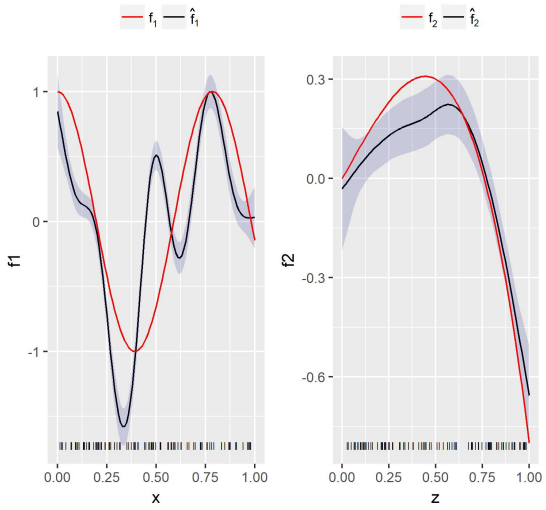


Figure 3.7: Non robust estimation of f_1 and f_2

We can clearly see, by looking at the left plot showing f_1 and its prediction $\hat{f}_1$, that the non-robust GAM approach fails. It results in a very bad and wiggly estimation of f_1, although only 8% of the dataset is corrupted.

Using the rgam package, which is an implementation of the methods described in [5] - see Section 3.2, we obtain the following solution plots:

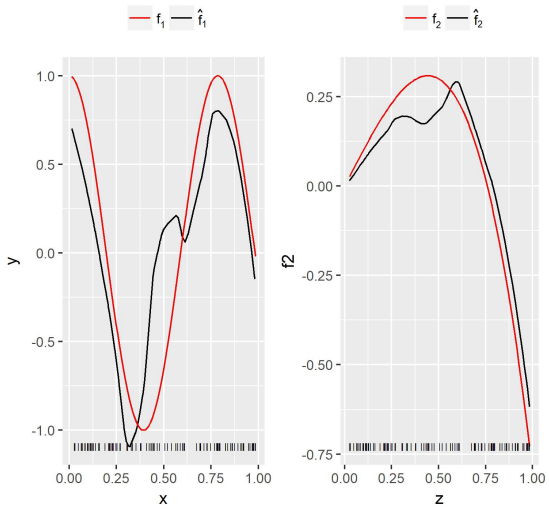


Figure 3.8: Robust estimation of f_1 and f_2

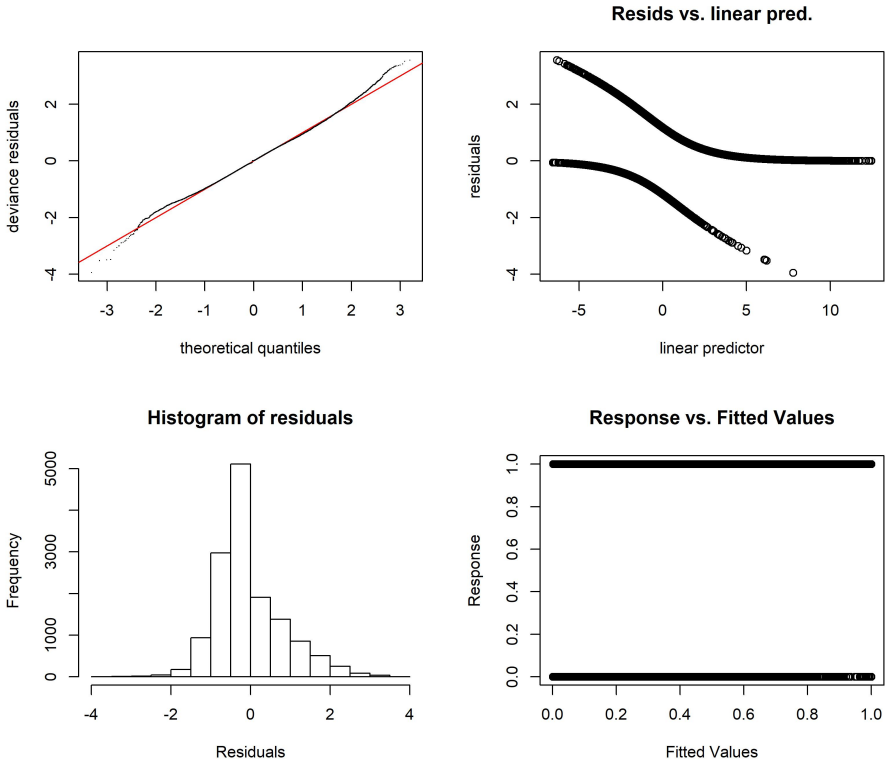
Although there is still a discrepancy between the true functions f_1 respectively f_2 and its estimate $\hat{f}_1$ and $\hat{f}_2$, we can see that, all in all, the estimates get very close to the truth - one mustn't forget that the model is also misspecified not considering interactions. Thus, the robust approach delivers, for this dataset, much better results than the non-robust approach; this is evident when we take a look at Figure 3.7 comparing it to Figure 3.8.

3.3.3 A real world example

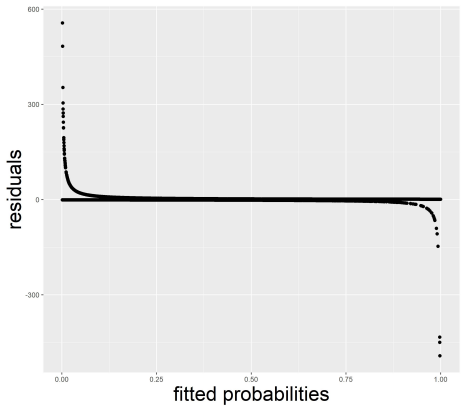
Finally, we will look at a "real" data example to illustrate the usefulness of GAMs. We use the "MAGIC Gamma Telescope" data set, which can be found at [9].
The data set consists of 19020 observations of eleven variables. The first ten are the, independent, continuous variables, $x_1, ..., x_{10}$, and the last one is the dependent categorical 0-1 variable y - representing an observed signal, encoded with g, or a background event, encoded with h. We model the relationship between y and $x_1, ..., x_{10}$ by a binomial distribution. Therefore, we use a so called Logit-link, which means that we model the mean of Y - or equivalently the probability $P(Y = 1)$ - by

$$\frac{exp(f_1(x_1) + ... + f_{10}(x_{10}))}{1 + exp(f_1(x_1) + ... + f_{10}(x_{10}))}.$$

Before calculating the fits $\hat{f}_1, ..., \hat{f}_{10}$ we randomly split the data set into a test and training set, where the test set contains around $\frac{1}{4}-$th of the original data. Using the mgcv package we obtain on this test set a total misclassification rate of 0.14 - where we classified new points according to the fitted probabilities with the threshold of 0.5 - and a probability of wrongly classifying a background event h as a signal of 0.16 - the latter is important for this data set. Furthermore we have the following residual plots:

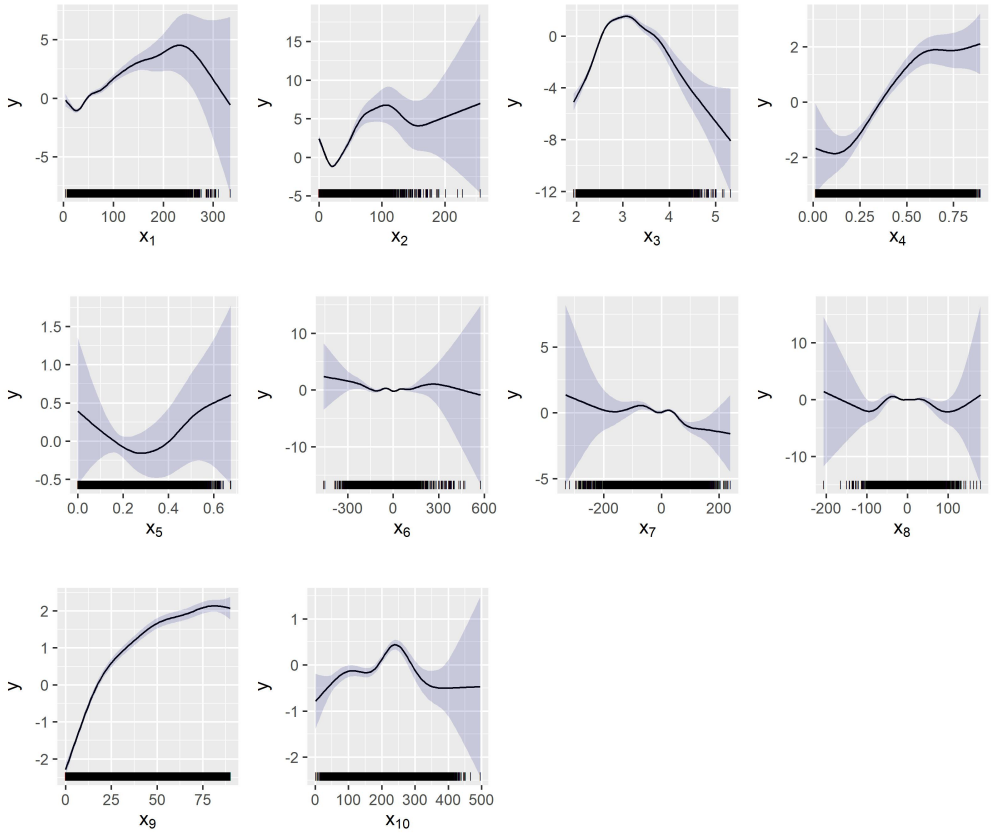


We can see that the deviance residuals are close to normal with mean zero. This might indicate that our model is adequate - as a deviance residual close to zero means that the fitted probabilities are also close to the observed ones. However, as can be seen from the linear predictor vs residuals plot higher and lower values tend to have bigger residuals. Examining this more closely by plotting fitted probabilities vs residuals we can see indeed that there might be a slight problem of having outliers at very low and very high probabilities:



Furthermore, as we have seen before, we have a misclassification rate of 0.14, on an independent test set. This leads to the conclusion that our model is indeed an adequate model. Further improvement might be achieved by considering multiple interactions.

At last, we show the plots for the obtained estimates $\hat{f}_1, ..., \hat{f}_{10}$ with their confidence bands - the ticks in the x-axis represent the observed values:



We can see that almost all the variables contribute in a very nonlinear way to the outcome - only x_6 and x_7 are closely linear. Also, for almost all the variables, we can see that the confidence intervals get very wide at the sides. This is usually due to less data there. Furthermore we can see that, in quantity, x_1, x_2, x_3, x_4 and x_9 explain the most of $P(Y = 1)$, whereas the others are either close to zero - like x_6, x_8 - or very small in comparison - x_5, x_7, x_{10}. We can see from the estimated functions in x_4 and x_9 that these seem to be, almost, non-decreasing functions. The behaviour of the estimates in x_1, x_2 and x_3 seem to be more irregular.

Summary

We started this thesis by considering the following. Having observed the values $(y_i, x_{i1}, ..., x_{ip})$, for $i = 1, ..., n$, we could try to model the relationship between these in the following way:

$$y_i = a_0 + a_1 x_{i1} + ... + a_p x_{ip} + \epsilon_i,$$

where ϵ_i are independent Gaussian distributed variables with mean zero and unknown constant variance. The goal is then to estimate the unknown parameters $a_0, ..., a_p$. Looking at it as a minimization problem, see (2.1), we followed a train of thought leading us to the so called least squares problem - see Chapter 2.

As the assumption of Y being a Gaussian random variable which is linearly depended on $X_1, ..., X_p$ doesn't always hold in reality, we determined that it is important to, on the one hand, also consider interactions of $X_1, ..., X_p$ and, on the other hand, to consider the case where Y is not Gaussian. However, as we have seen, we explained that there are two main problems. Firstly, considering too many interactions usually leads to a computationally infeasible problem and, secondly, is not clear at the beginning how to generalize such a model to Y not being Gaussian.

We thus went on to remedy these two problems individually at first. At the beginning of Chapter 2 we considered additive models in order to keep the exploding cost when considering interactions small. At the beginning, we presented the Hilbert space approach to AMs, which is mainly taken in [1], but then decided to continue with a penalized approach as it is taken in [2]. This led us to a, loosely speaking, nonparametric penalized least squares problem - see (2.20). As part of this we introduced and motivated smoothing splines.

To find a remedy to the limitation of Y not being normally distributed, with constant variance, we introduced generalized linear models. For the latter we

saw that a likelihood approach, see (2.30), to estimate the parameters makes more sense than a least squares approach.
Trying to solve the likelihood problem then led us to the IRLS algorithm, which enables us to get a concrete estimate of the parameters.

In the third Chapter we then combined these two approaches to find a remedy for the two problems at the same time and chose to take again a likelihood approach, as it is heuristically better motivated, but this time adding a penalty to it. Here, yet again, the usefulness of splines proved itself and this approach led to (3.1). Solving such a penalized likelihood problem, which luckily turned out to be parametric after all, led to the P-IRLS algorithm. As we have seen, the P-IRLS is basically an extension of the IRLS algorithm, with the limitation that in the case of Y not being Gaussian a lot of caution needs to be paid to whether it converges, a solution is unique, etc - see Theorem 3.1.
We then went on to discuss some big sample limit results and the generalization of some definitions and methods, for smoothing parameter selection, from GLMs to GAMs.

Before ending with some examples illustrating the explored models and methods we discussed one possible way to robustify the GAM framework, so that in the presence of outliers or corrupted data our methods do not deliver "incorrect" estimations of the parameters, see Section 3.2.

Finally, we used two simulated and one real data set example to show the methods so far discussed and to also reveal where difficulties in the non-outlier and outlier cases lie; see Section 3.3. We saw that a robustification of the GAM methods is absolutely necessary. Without any sort of robustification the usual methods delivered vastly wrong results - with a corruption of under 10% in the simulated data cases.

There are still many topics concerning GAMs which we have not covered in this thesis. For example, we could further explore vector GAMs, so called VGAMs - meaning that Y is not necessarily one dimensional anymore - see [7] for example.
Furthermore, we could also explore extensions of GAMs which treat their inflexibility concerning structural breaks better - as they do not cope too well with the case where the true underlying functions are piecewise discontinuous functions.
Another topic to cover could also be the mixture of random variables belonging to the exponential family and how GAMs could be extended to such.

At last, but surely not least, the mixture of GAMs and other models, such as trees for example, could be explored.

Bibliography

[1] T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models* Monographs on Statistics and Applied Probability 43, Chapman & Hall, 2-6 Boundary Row, London SE1 8HN 1990

[2] S. N. Wood, *Generalized Additive Models An Introduction with R.* Texts in Statistical Science, Chapman & Hall/CRC, Taylor & Francis Group, New York, 2006

[3] S. N. Wood, *Generalized Additive Models An Introduction with R Second Edition.* Texts in Statistical Science, Chapman & Hall/CRC, Taylor & Francis Group, New York, 2017

[4] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning.* Springer Series in Statistics Springer New York Inc., New York, NY, USA, 2001

[5] A. Alimadad and M. Salibian-Barrera, *An outlier-robust fit for Generalized Additive Models with applications to disease outbreak detection.* Journal of the American Statistical Association, Vol. 106, No. 494 (June 2011), pp. 719-731

[6] G. Boente, A. Martinez and M. Salibian–Barrera, *Robust estimators for additive models using backfitting.* Journal of Nonparametric Statistics, Volume 29, 2017 - Issue 4, pp. 744-767

[7] T. Yee, *Vector Generalized Linear and Additive Models.* Springer Series in Statistics, Springer-Verlag New York, 2015

[8] A. Dobson, *An Introduction to Generalized Linear Models, Third Edition.* Texts in Statistical Science, Chapman & Hall/CRC, Taylor & Francis Group, 2008

[9] R. K. Bock, *Major Atmospheric Gamma Imaging Cherenkov Telescope project (MAGIC)* and P. Savicky. UCI Machine Learning Repository

[https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope]. Irvine, CA: University of California, School of Information and Computer Science

Curriculum Vitae

Name:	Rieser Christopher
Date of birth:	19.12.1988
Place of birth:	Schwaz, Austria
till 2007:	Lycée International de Saint-Germain-en-Laye, France
2008-2013:	TU Graz, Austria, Bachelors Applied Mathematics
2013-2016:	University of Vienna, Austria, Masters Mathematics
2017-2018:	TU Vienna, Austria, Masters Statistics