Fitting Generalized Additive Models for very large datasets with Apache Spark

Chapter Excerpt

Kai Thomas Brusch

24.12.2015

Summary This document contains three introductory chapters for my bachelor thesis titled "Fitting General Additive Models for very large datasets with Apache Spark".

Contact kai.brusch@gmail.com

Location Hochschule für Angewandte Wissenschaften Hamburg

Department Dept. Informatik

Examiner Prof. Dr. Michael Köhler-Bußmeier

Second examiner Dipl.-Math. Markus Schmaus

Contents

1 Linear models

1.1 Introduction to linear models

Linear models are statistical models in which an univariate response is modeled as the sum of a 'linear predictor' and a zero mean random error term. The linear predictor depends on some predictor variables y, measured with the response variable x, and some unknown parameters β plus an error term ϵ, which must be estimated. This process is formally stated for a given row i of data as: [Wood, 2006] [Wood et al., 2015] [Zaharia et al., 2010]

$$y_{\mathrm{i}} = \beta x_{\mathrm{i}} + \varepsilon_{\mathrm{i}} \tag{1}$$

The are many choices for β and finding the best possible β stands at the heart of the following chapter. A key feature of linear models is that the linear predictor depends linearly on these parameters. Statistical inference with such models is usually based on the assumption that the response variable has a normal distribution. Linear models are used widely in most branches of science and an example is given in the next section.

1.2 Example of a linear model

Before describing the relevant theory I would like to give an example of a simple linear model. Lets say we have a data set called mtcars thats describes cars with respect to miles per galon (mpg) and horse power(hp). After careful thought and the examination of a scatter plot I believe that there is a linear relation ship between the the miles per galon and the horse power of a car. I also believe that miles per gallon follow normal distribution. I would try to explain the relationship between hp and mpg as a linear model takes miles per galon as the dependant variable and horse power as the independant variable. This model decription written in R yields the following line of code:

```
model <- lm(data=mtcars,hp ~ mpg)

summary(model)
```

The summary gives me the estimated model and we can see the slope and the insect of function explaining the relationship between dependant and independant variable:

```
Call:
lm(formula = hp ~ mpg, data = mtcars)

Coefficients:
(Intercept)          mpg
    297.688       -8.022
```

Our model describes hp power as $-8.022 * mpg + 297.688$. We can see that the more hp a car has the fewer mpg does it offer. The red line in the plot is our estimated function that illustrates the estimated model

The natural question that arises is: how to we estimate the line and the error of the chart above? The relevant method is called ordinary least squares and will be introduced in the next section.

1.3 Ordinary least square estimation of β

We are now looking at methods of finding β. Ideally we want to choose a β that produces a line through our data points with minimal distance between our points and our estimated line.

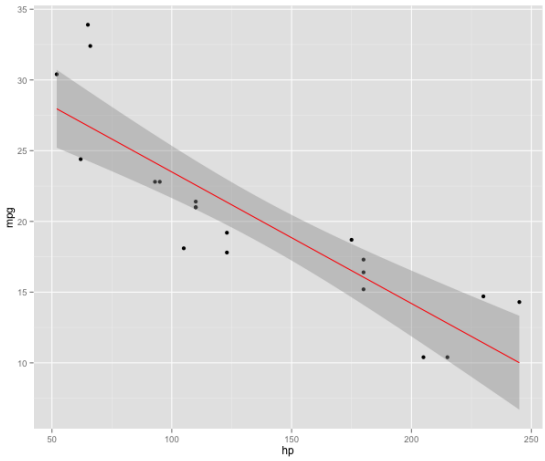

Figure 1: $-8.022 * mpg + \ 297.688$

More precicely we are looking to estimate a β that minimizes the squared distance between an estimated β times the given x and y. We are squaring the distance to normalize negative and positive differences. This distance formaly describes as S:

$$\mathbf{S} = \sum_{i=1}^{n} (y_{\mathrm{i}} - x_{\mathrm{i}}\beta)^2 \tag{2}$$

The close our S gets to 0 the better our line fits the data. The Markov-Gauss Theorem states that the minimization of S yields $\widehat{\beta}$ which is the best possible estimation for β. This shall be discussed n the next chapter. 1 represents the univariate case where y is explained with only one variable. The process of minimizing S is commonly refered to as ordinary least squares (OLS).

There are two ways to think about estimating β, thinking off OLS in terms of calulus on functions allows us to compute $\widehat{\beta}$ for the univariate case while computing $\widehat{\beta}$ for multiple independant variables requires linear algebra. From the calculus perspective we can see OLS as function with two parameters: S and β. Minimizing S equates to taking the partial derivative of S with repsect to β. This is the most common approach and offers insight by stating S as the following equation:

$$\frac{\partial S}{\partial \beta} = -\sum_{i=1}^{n} 2x_i(y_i - x_i\beta) \tag{3}$$

Rewriting the partial dereviative to yiel $\widehat{\beta}$ gives a very good idea of β

$$-\sum_{i=1}^{n} 2x_i(y_i - x_i\widehat{\beta}) = 0 \tag{4}$$

$$-\sum_{i=1}^{n} x_i y_i - \widehat{\beta}\sum_{i=1}^{n} x_i^2 = 0 \tag{5}$$

$$\widehat{\beta} = \sum_{i=1}^{n} x_i y_i / \sum_{i=1}^{n} x_i^2 \tag{6}$$

Minimizing S w.r.t. β to compute $\widehat{\beta}$ is a reasonable approach when dealing with one independant variable. However, almost all relevant applications involve much more than one independant

variable and require linear algebra. To estimate $\widehat{\beta}$ with OLS for multiple independant variables involves rephrasing the questions in terms of linear algebra. First we have to state the process of finding β as a linear combination problem whichs equates to asking what linear combination of column vectors of our model matrix X and our vector β of unknown coefficents yields the vector of y. Formally:

$$X\beta = y \tag{7}$$

Now that we have restated OLS as a matrix problem we can apply some linear algebra to find $\widehat{\beta}$. First we can restate the problem of finding β as finding the unkown vector β equates to finding a linear combination of our column vectors of X with β that result in y. Finding this linear combination is highly dependant on the properties of X. Taking a look at relevant model matricies X will show that they almost exclusively consist of m rows and n columns with $m > n$. Being $m > n$ implies that the matrix is not symetric and not invertible. Given that X has more rows than columns we can think of $X\beta = y$ as a system of equations with more equations than variables which causes this sytem of equation to have no solution. We have to stress the fact that the sytem of equations of 7 does not have a solution because there is no possible selecton for β that lies in the column vector space of X. Whilst the nature of X makes finding β impossible we can find an estimation $\widehat{\beta}$ of β by projecting it back into the column space of X. The solution becomes to multiply by A^TThe proper projection p is defined as $p = X\widehat{\beta}$. The process of projecting of finding the projection involves multiplying by the transpose of the model matrix yielding:

$$X^T X\widehat{\beta} = X^T y \tag{8}$$

This projection comes however at the cost of an error term:

$$y = X\widehat{\beta} + \epsilon \tag{9}$$

$$\epsilon = y - X\widehat{\beta} \tag{10}$$

The questions that now arises is: how to we make $\epsilon = y - X\widehat{\beta}$ as small as possible? Using algebra we can split the vector β into two parts. One part in the column space is our projection p and the perpendiclar part in the nullspace of A^T which the ϵ. It is essential to remember that the column space is always perpendicular to the nullspace of A^T. The solution to $X\beta = p$ leaves the least possible error ϵ, returning to the previously stated method of least squares, but this time in the world of linear algebra:

$$\|X\beta - y\|^2 = \|X\beta - p\|^2 + \|\epsilon\|^2 \tag{11}$$

This is the law $c^2 = a^2 + b^2$ for a right angle. The vector $X\beta - p$ in the column space is perpendicular to ϵ in the nullspace of A^T. We reduce $X\beta - p$ to zero choosing β to be $\widehat{\beta}$, this leaves us with the smallest possible error vector ϵ. The projection leaves us with an invertible matrix that can be solved by usual elimination. The least squares solution $\widehat{\beta}$ makes $\epsilon = X\beta$ as small as possible.

1.4 Gauss-Markov Theorem

So far we have introduced two different methods on how to estimate β. The calculus way for univariate data and the linear algebra way for multivariate data, both estimated β with an approximate $\widehat{beta}$ which seeks to minimize the squared error of actual and estimate. While squared error seems like the an obvious choice we have no formal reason to prefer it to other measures . This section will lay out the formal proof that shows why $\widehat{\beta}$ is the best possible estimator.[Wood, 2006] states that there exists no estimator with lower variance than least squares estimation. The

1.5 Estimating $\widehat{\beta}$ with orthogonal decomposition

The previously suggested method lends a great tool to think about the univariate least squares technique. But the suggested methods is rarely applied Any m by n matrix X with independant columns can be factored into QR. The m by n matrix Q has orthonormal columns and the square matrix R is upper triangular with positive diagonal. X^TX equals $R^TQ^TQR = R^TR$ simplifying the leas squares equation to $Rx = Q^T\beta$; allowing us to multiply by R^T instead of X^T for form the square matrix. This additional simplicity allows us to restate the problem of least squares for matrices using QR decomposition as:

$$R^TR\widehat{\beta} = R^TQ^Ty \; or \; R\widehat{\beta} = Q^Ty \; or \; \widehat{\beta} = R^{-1}Q^Ty \tag{12}$$

A very important property of orthogonal matrices is that their multiplication with a vector is ideponent with respect to the lenght of the vector.

Instead of multipliying with a fully formed model matrix X we only multiply with small subset, thus reducing the amount of mulipilications and memory requirements drasticaly.

The QR decomposition is formally stated in the appendix and will appear as an essential part later in this thesis.

2 Generalized Linear Models

2.1 Introduction to Generalized Linear Models

[**?**] describes Generalized Linear Models (GLMs) as an extension of the general linear model with the ability to model more expotential family response disributions. While linear models with OLS estimation only allow for a normal distributed response, GLMs allows to model the reponse variable from any arbitrary expotential family. The expotential family of distributions contains many pratical useful distibutions. A formal description of it's basis structure can be given as

$$g(\mu_i) = X_i\beta_i \tag{13}$$

Where $\mu_i \equiv E(Y_i)$ and Y_i is distibuted according to some expotential family. Its members include but are not limited to Poisson, Bionomial, Gamma and Normal Distributions. Every expotential family distribution has a link function g().The idea here is that linear functions of the predictor variables are obtained by transforming the right side of the equation (f(x)) by a link function. The data are then fit in this transformed scale (using an iterative routine based on least squares), but the expected variance is calculated on the original scale of the predictor variables. X_i is the i^th row of a model matrix X and β is a vector of unknowns parameters. The link functions allows us to model an expotential function ins terms of a linear link function. To account for expotential familty distributions comes at a cost however: While OLS was sufficient for estimating β for normal distributed data now have to generalize this notion to acount for an arbitrary amount of distribution parameters. The generalization of OLS is called maximum likelihood estimation (MLE) and involves an iterative method called iterative re-weighted least squares (IRLS). An important practical feature of generalized linear models is that they can all be fit to data using IRLS, independant of response variable distribution. A further layer of complexity is the error term now has varying variance, our estimations and inference needs weight the likelihood accordingly.

2.2 Example of a Generalized Linear Models

Examples for generalized linear models are found in many fields of science. Allowing the reponse to be distributed to poisson or binomial dsitributions allow to model many pracitical aplications. Let us consider an example of a poisson distributed reponse. Poisson distributions are most

commonly used to model the number of occurences of a given event. Lets say we want to discover the relationship between smoking tabaco und number of death for a given age group drawn from a random population. We are given a data set that desribes a random sample in terms of age range, deaths and a categorial desription of the smoking habit. The age groups are non-overlaping five year spans starting at 40. The smoking habit is categoriesed as ("no", "cigarPipeOnly", "cigarrettePlus", "cigarretteOnly") We can use the given data to produce a model that models the number of deaths as a function of smoking habit and age group. The model we are trying to build can be best described with the following R code snipped:

```
Call:  glm(formula = dead ~ +0 + as.factor(smoke) + as.factor(age),
family = poisson(link = "log"), data = data)

Coefficients:
as.factor(smoke)cigarPipeOnly   as.factor(smoke)cigarretteOnly
                        3.5088                          4.6131
as.factor(smoke)cigarrettePlus                  as.factor(smoke)no
                        4.8864                          3.2702
        as.factor(age)45-59             as.factor(age)50-54
                        0.1340                          0.3163
        as.factor(age)55-59             as.factor(age)60-64
                        1.3956                          1.9252
        as.factor(age)65-69             as.factor(age)70-74
                        1.8867                          1.5894
        as.factor(age)75-79               as.factor(age)80+
                        1.1377                          0.7576
```

The estimated coefficients paint a clear picture: Less deaths occur in non-smoking groups.A little more for cigar pipe only smokers and many more for cigaretter smokers, indeferent to frequency. However, the coefficent interpretation should be secondary in the example and focus should on the model description. A Poisson distribution need only one parameter thus we only supply the number of death occured. The have to apply the link function in the right side of our equation, yielding in the following model description $Poisson(dead) = Constant + log(smoke) + log(age)$. We can see that used a GLM model with death as the reponse and specified the expotential family and its link. Taking a look at the variance of the

2.3 Maximum likelihood estimation

Parameter estimation stands at the heart of all statistical models. Random variables and their distribution allows us to explain a variables behavior based on their parameters. Once a model is specified with its parameters, and data have been collected, one is in a position to evaluate its goodness of fit, that is, how well it fits the observed data. Goodness of fit is assessed by finding parameter values of a model that best fits the data procedure called parameter estimation. The OLS linear model estimated the μ and σ for $N(\mu, \sigma^2)$ by minimizing S. [J. A. Nelder, 1972] introduces a method to account estimate parameters for any expotential family distributions members. Expotential family distributions have diffent number of parameters and parameter meaning, cf. $P(\lambda)$ $G(K, \theta)$. Maximum likelihood estimation provides a single framework to allow parameter estimation for any of the expotential family distributions.

From a statistical analysis point of view the vector y of observed data is a random sample from an unknown population. The goal of maximum likelihood estimation is to find the parameters of the given distribution that most likely has produced this sample. This process is discribed with a probability density function (PDF) $f()$ of observed data y given a parameter w: $f(y|w)$. If individual observations, y_i's, are statistically independent of one another, then according to the

theory of probability, the PDF for the data y given the parameter vector w can be expressed as a multiplication of PDFs for individual observations.

$$f(y|w) = f((y_1, y_2, ..., y_n)|(w_1, w_2, ...w_n)) = \prod_{i=1}^{n} f_i(y_i|w_i)) \tag{14}$$

Given a set of parameter values, the corresponding PDF will show that some data are more probable than other data. In reality the data is already given and we are searching for the parameters of the distribution that mostlikely produced the data. We thus inverse our function $f(y|w)$ to $L(w|y)$ to produce the likelihood of y given the parameters w. The principle of maximum likelihood estimation, originally developed by R.A. Fisher in the 1920s, states that the desired probability distribution is the one that makes the observed data "most likely," which means that one must seek the value of the parameter vector that maximizes the likelihood function $L(w|y)$: The resulting parameter vector, which is sought by searching the multi-dimensional parameter space, is called the MLE estimate.

Theoretically MLE estimates need not exist nor be unique. But if they exist and are unique they can be estimated. For computational convenience, the MLE estimate is obtained by maximizing the log-likelihood function, $ln(L(w|y))$: This is because the two functions, $ln(L(w|y))$ and $L(w|y)$; are monotonically related to each other so the same MLE estimate is obtained by maximizing either one and the log-likelihood is prefered for obvious reasons. Assuming that the log-likelihood function, $ln(L(w|y))$ is differentiable, if wMLE exists, it must satisfy the following partial differential equation known as the likelihood equation:

$$\frac{\partial lnL(w|y)}{\partial w_i} = 0 \tag{15}$$

This is because the definition of maximum or minimum of a continuous differentiable function implies that its first derivatives vanish at such points. The likelihood equation represents a necessary condition for the existence of an MLE estimate. An additional condition must also be satisfied to ensure that ln $ln(L(w|y))$ is a maximum and not a minimum, since the first derivative cannot reveal this. To be a maximum, the shape of the log-likelihood function should beconvex (it must represent a peak, not a valley) in the neighborhood of wMLE: This can be checked by calculating the second derivatives of the log-likelihoodsand showing whether they are all negative.

$$\frac{\partial^2 lnL(w|y)}{\partial w_i^2} < 0 \tag{16}$$

In practice, however, it is usually not possible to obtain an analytic form solution for the MLE estimate, especially when the model involves many parameters and its PDF is highly non-linear. In such situations, the MLE estimate must be sought numerically using nonlinear optimization algorithms. The basic idea of nonlinear optimization is to quickly find optimal parametersthat maximize the log-likelihood. This is done by searching much smaller sub-sets of the multi-dimensional parameter space rather than exhaustively searching the whole parameter space, which becomes intractable as the number of parameters increases. The "intelligent" search proceeds by trial and error over the course of a series of iterative steps. Specifically, on eachiteration, by taking into account the results from the previous iteration, a new set of parameter values is obtained by adding small changes to the previous parameters in such a way that the new parameters are likely to lead to improved performance. Different optimization algorithms differ in how this updating routine is conducted. This process is called

2.4 Fitting Generalized Linear Model

Before looking at the fitting process of GLM it is important linear models assume an error with constant variance and no covariance, the combination of the two conditions is called homoscedastic. The assumption of homoscedastic is now dropped, in GLM the error term has a varying variance and error can have covariance. This situation is called heteroscedastic. Our fitting method thus needs to account for the varying error. As we have discussed previously, we obtain MLEs by setting the score vector equal to 0 Recall that for a GLM using the canonical link function, the score vector is:

$$u(\beta) = \phi^{-1} X^T (y - \mu) \tag{17}$$

In the above equation μ is a function of $\eta = X\beta$. However this is not required to be a linear function, if this is not the case lack a closed-formed sulution for β. If we are given the canoical link function g we may use a Taylor series approach to obtain the following approxiamtion of the point $\widehat{\beta}$:

$$\mu = \widehat{\mu} + W(X\beta - X\widehat{\beta}) \tag{18}$$

Where $\widehat{\mu} = g^{-1}(X\beta)$ when the g is defined as

$$\frac{d}{d\eta} g^{-1}(\eta) = W(\eta) \tag{19}$$

Thus we can obtain the following linear approximation to the score for a given β:

$$\frac{\partial l}{\partial \beta} \approx \phi^{-1} X^T W (z - X\beta) \tag{20}$$

where $z = X\widehat{\beta} + W^{-1}(y - \widehat{\mu})$ is known as the adjusted response. Note that this approximation is based at $\widehat{\beta}$ or, equivalently, $\widehat{\mu}$, which are treated as constants in the above expression, thereby rendering the score equation linear in β after the approximation Again, recall that this approximation will be accurate near the fitted values $\widehat{\mu}$ but not necessarily accurate far away from them. As we saw previously this gives the max likelih hood estimate:

$$\widehat{\beta}^{(m)} = (X^T W X)^{-1} X^t W z \tag{21}$$

Note that W here plays the role of the weights in weighted least squares, and for that reason is often referred to as the weight matrix.Again, recall that for the canonical link, W is entirely determined by the mean-variance relationship, and that it plays a prominent role in the variability of $\widehat{\beta}$ as well.Note that in the above equation, we require a superscript on $\widehat{\beta}^{(m)}$ because this sia case of unknown weights, where W (and z) will change depending on $\widehat{\beta}$ and vice versa

As we saw earlier, one way to address this problem is to iterate the process of reweight–estimate–reweight–estimate–. . . until convergence; this iteratively reweighted least squares (IRLS) algorithm is how generalized linear models are fit:

1. Choose an intial value $\widehat{\beta}^{(0)}$
2. For m = 0,1,2,...
   - 2.1. Calculate z and W based on $\widehat{\beta}^{(m)}$
   - 2.2. Solev for $\widehat{\beta}^{(m+1)}$
   - 2.3. Check to see whether $\widehat{\beta}^{(0)}$ has converged: if yes then stop

This concept it generalized with Newton-Raphson

3 Generalized Additive Models

3.1 Introduction to Generalized Additive Models

Generalized Additive Models (GAMs) were first introduced by [Hastie, 1990]. Hastie and Tibshirani established a simple yet powerful model family that relies on a subtle model: Relationships between the predictors and the dependent variable follow smooth patterns that can be linear or nonlinear. The smooth relationship can estiated simultatiously and then added up. Formally am GAM is specified by the linear prediction in terms of the summation of smooth functions. This allows for a more flexible modeling of the influence for each explanatory variable.

$$g(E(Y)) = \mathbf{X}_{\mathrm{i}}\Theta + f_1(x_{1\mathrm{i}}) + f_2(x_{2\mathrm{i}}) + f_3(x_{3\mathrm{i}}, x_{4\mathrm{i}})... \tag{22}$$

Y is the dependant variable that we are trying to predict. Since we are modelling the expected value of Y we write $E(Y)$, the $g()$ denotes the link function that links the expected value to the precitions x_i, each $f_i()$ is smooth function for that x_i. Finding the right smooth function stands at the heart of GAM fitting and will be discused in the next section.

3.2 Smooth Functions

[Hastie, 1990] emphasizes the importance of smooth functions and describes them as 'the cornerstone of Generalied Additve Models'. Smooth function for GAMs are divided into three categories:

- Local Regression
- Smoothing Splines
- Regression Splines (B-Splines, P-Splines, thin plate splines, cubic splines)

Local regression covers a range of nearest neighboorhood-based smoother which provide a good introduction to smoothers with little practicability. Smoothig splines use the penelized sum of squares with similar to GLMs. Regression splines are the most practical. They are computationally cheap, and can be written as linear combinations of basis functions that do not depend on the dependent variable, Y , which is convenient for prediction and estimation.

3.2.1 Local Regression

Loess belongs to the class of nearest neighborhood-based smoothers. Loess can be , we have to understand the most simplistic member of this family: the running mean smoother. Running mean smoothers are symmetric, moving averages. Smoothing is achieved by sliding a window based on the nearest neighbors across the data, and computing the average of Y at each step. The level of smoothness is determined by the width of the window. While appealing due to their simplicity, running mean smoothers have two major issues: they're not very smooth and they perform poorly at the boundaries of the data. This is a problem when we build predictive models, and hence we need more sophisticated choices, such as loess. Loess are a natural extension of the running mean smoother. Loess produces a smoother curve by fitting a weighted regression within each nearest-neighbor window, where the weights are based on a kernel that penelizes data points far from the current mean. [**?**] provides an illustrative example of loess vs running mean. [**?**] determines smoothness using a span parameter. The span parameter determines how much of the data the sliding mean will account for, a symetric span of 0.6 will account for 0.3 of the data the right and left of each observation. Calculate $d_i = (x_i x)/h$ where h is the width of the neighborhood. Create weights using the tri-cube function $wi = (1d_i^3)^3$, if x_i is inside the neighborhood, and 0 elsewhere. Fit a weighted regression with Y as the dependent variable using the weights from step 3. The

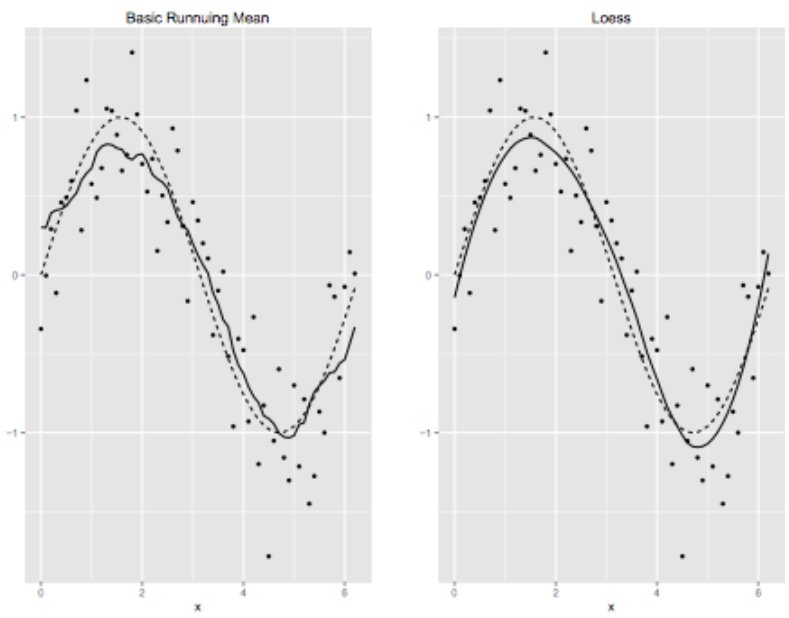

Figure 2: Basic running mean vs Loesss

fitted value at target data point x is the smoothed value. The following example illustrates both methods for smooth finding on data following a general S pattern:

2 Illustrates the differences in fit between loess and running average. It is obvious that loess provides a smoother fit of the data. While nearest neighborhood-based smoother provide fast results with minimum theory they suffer two fatal flaws: We can't control the smoothness and the methods lacks stability with little neighboors, for example on the edges of the plot.

3.2.2 Smoothing Splines

Smoothing splines provide a very different method to smooth finding. Other than nearest-neighboorhood methods smoothing splines estimate smooths by minimizing the penalized squared error.

$$\sum_{i=1}^{n}(f(x)-x)^2+\lambda\int(s''(x))^2dx \tag{23}$$

Where the squared error is identical to the measure S from linear models.

$$\sum_{i=1}^{n}(f(x)-x)^2 \tag{24}$$

With the addition of a penalty term . λ is a penalty factor that controls the smoothness by scaling the second derivative of the smooth function, intuitivly is penalized wiggliness.

The integrated square root of second derivative penalizes models that have a too 'rough' function. As λ approaches 0 f will become a straight regression line while λ of 0 becomes an unpenelized

regression spline. The trade off between model fit and and model smoothness is controlled by the smoothing parameter λ. The linear nature of f allows us to rewrite 27 in a more concise form.

$$\lambda \int (s''(x))^2 dx \tag{25}$$

3.2.3 Regression Splines

Smoothing splines

GAMs are fomally described by the following equation:

22 explains y_i as the model matrix for this row and the smooth functions $f_j(x_1 j)$ of the x values for this row. X_i is a row of the model matrix with parametric component θ. Unlike the linear model we can now specify a smooth function for each explanatory variable. This proves to be way more flexible than only allowing for a constant influence per explanatory variable. The natural question that arises now are: How do I find proper smoothing functions? Finding the right smooth function stands at the heart of GAM fitting and can be best illustrated in the univariate case 23.

$$y_i = f(x_i) + \epsilon_i \tag{26}$$

Smooth functions form a vector space, which can be approximated using a linear basis. Only allowing linear basis allow us to heavily leverage the theory already developed for linear models and S as the optimal model fit. For the sake of illustration we assume that 23 can be rewriten as the following equation if $b_i(x)$ is the ith basis function:

$$f(x) = \sum_{i=1}^{q} b_{\mathrm{i}}(x)\beta_{\mathrm{i}} \tag{27}$$

In 24 we already know $f()$ is linear in regard to 24. We now have to specify a basis function to represent bi. We can choose from many basis functions for bi, each with advantages and disadvantages. A common choice however is a fourth order polynomial basis function. 24 represented by a fourth order polynomial yields the following model:

$$f(x) = \beta_1 + x\beta_2 + x^2\beta_3 + x^3\beta_4 + x^4\beta_5 \tag{28}$$

Applying 25 to 23 we get the modeling of y_i as the sum of smoothing functions.

$$y_i = \beta_1 + x_i\beta_2 + x_i^2\beta_3 + x_i^3\beta_4 + x_i^4\beta_5 + \epsilon_i \tag{29}$$

3.3 Regression Splines

Equation 26 gives a formal full description of $\mathrm{y_i}$ as the sum of a fourth degree polynomial. Fourth order polynomial basis does a fine job of illustrating the used concepts but enforce rather strict limitations. Instead of trying to fit a single fourth order polynomial to all our data points we will now try to represent f with cubic splines. This approach is called regression splines and follows this general pattern:

- Divide the curve in to a fixed number of sections, the point between two sections is called a knot.
- Find a cubic polynomial for each section between two knots
- Each cubic polynomial must match it's neighbors in value, first and second derivative at the location of the knot.

Using cubic splines as a basis for f means that 23 becomes a linear model identical to **??**. We can hence rely well established linear model theory to fit a good polynmial. The number and location of knots will influence the fit of our splines and are carefully chosen. The right number and location of knots can be estimated but are usually a problem speicifc design decision. After establishing how to represent the smooth functions and the use of knots we will now turn to the smoothing parameter λ. λ is a penalty factor that controls the smoothness by scaling the second derivative of the smooth function, this can be formalized by:

$$\|y - X\beta\|^2 + \lambda \int_0^1 [f''(x)]^2 dx \tag{30}$$

The integrated square root of second derivative penalizes models that have a too 'rough' function. As λ approaches 0 f will become a straight regression line while λ of 0 becomes an unpenelized regression spline. The trade off between model fit and and model smoothness is controlled by the smoothing parameter λ. The linear nature of f allows us to rewrite 27 in a more concise form.

$$\|y - X\beta\|^2 + \lambda \beta^T S \beta \tag{31}$$

28 shows that describing an unknown variable as the sum of smooth functions can be restated as the minimization of penalized regression splines and hence becomes problem of minimizing equation 28. While the the problem of finding a smooth function for a univariate function has almost been solved by 28 we are still left with an unknown parameter of λ. Finding the optimal λ is subject of the following section. It is important to remember that this λ is not the distribution parameter from MLE but the smoothing parameter λ for GAMs.

3.4 Smoothing parameter Estimation

Finding a good smoothing parameter through minimizing 28 the essential question for GAM research. Finding an optimal choice for λ means to trade goodness of fit to smoothness. A high λ will overfit the data while a low λ will create undersmoothed splines. The common approach is to define a measure for the predictive capabilities of our estimated spline. For our purposes we will use the generalized cross validation (GCV) which formalizes the notion that an ideal splines minimizes the average distance between our splines and the function f.Another interpretation is to see the GCV as the prediction error and that finding the best smoothing parameter is equivialent to find the smoothing parameter with the minimal prediction error. A formal definition of the GCV score yields:

$$V_g(\lambda) = \frac{n \left\| y - X\widehat{\beta}_\lambda \right\|^2}{\{n - tr(F_\lambda)\}^2} \tag{32}$$

29 formalizes that the generalized cross validation score for a given λ, formaly $V_g(\lambda)$ is n times the squared distance for a currently estimated parameter $\widehat{\beta}$ given a λ. This distance get put in perspective to degrees of freedom for the current λ: $\{n - tr(F_\lambda\}^2$. Hence the GCV score expresses goodness of fit as the tradeoff between fitting the points and degrees of freedom for a given λ. Finding λ is done via penalized likelihood iteratively reweighted least squares (PIRLS) which proceeds as follows, where V is the function such that $var(y_i) = \phi V(\mu_i)$ and $\mu_i = E(y_i)$. First initialize $\widehat{\mu}_i = y_i + \epsilon_i$ and $\eta_i = g(\mu_i)$ where ϵ_i is a small quantity (often 0) added to ensure that $g(\mu_i)$ exists.

Then iterate the following steps to convergence.

1. form $z_i = g'(\widehat{\mu}_i)(y_i \widehat{\mu}_i) + \eta_i$ and $w_i = V(\widehat{\mu}_i)^{1/2} g'(\widehat{\mu}_i)^{-1}$.

2. putting the wi in a diagonal matrix W, minimize the weighted version of expression 28:

$$\|Wz - WX\beta\|^2 + \lambda\beta^T S\beta \tag{33}$$

with respect to β to obtain $\widehat{\beta_i}$ and the updates $\widehat{\eta_i} = X\widehat{\beta}$, and $\widehat{\mu_i} = g^-1(\widehat{\eta_i})$.

For moderate size data PIRLS will converge for each trial λ. Iterating 30 to obtain the new estimates can place a heavy burden on the memory requirements. Each step of the PIRLS requires to form the whole model matrix X. The difficulty is simply that the model matrix for the model can become too big: if n and p are respectively the number of rows and columns of the model matrix, and M is the number of smoothing parameters, then the memory requirements of GAM fitting methods are typically $O(Mnp^2)$..The aim of my bachelor thesis is to discover and implement the in [Wood et al., 2015] suggested optimizations for PIRLS.Though I am able to reproduce the wording of this method I currently do not understand this method in detail, understading this method will be one of the core tasks in the next weeks, table 1 for reference.

4 Matrix Algebra

4.1 Orthogonal Matrices

We are concerned with matrices that do have orthogonal column vectors, severly concerned

4.2 QR decomposition

We want to decompose a matrix A in two parts, one orthonormal and one upper triangular, X = QR

5 References

[Hastie, 1990] Hastie, T. (1990). *Generalized additive models*. Chapman and Hall, London New York.

[J. A. Nelder, 1972] J. A. Nelder, R. W. M. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384.

[Wood, 2006] Wood, S. (2006). *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC.

[Wood et al., 2015] Wood, S. N., Goude, Y., and Shaw, S. (2015). Generalized additive models for large data sets. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64(1):139–155.

[Zaharia et al., 2010] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA. USENIX Association.