MASTER'S THESIS

Ismo Hannula

Statistical inference for generalized additive models with an application to mothers' depression symptoms

UNIVERSITY OF TAMPERE
School of Information Sciences
Statistics
May 2011

University of Tampere

School of Information Sciences

HANNULA, ISMO: Statistical inference for generalized additive models with an application to mothers' depression symptoms

Master's thesis, 44 pages, 10 appendix pages

Statistics

May 2011

Abstract

In statistics, linear modelling techniques are widely used methods to explain one variable by others. Generalized additive model, GAM, has developed from both generalized linear models, GLM, and nonparametric regression methods. This master's thesis aims to provide a solid background view on both of these two, eventually leading to generalized additive model.

Work on generalized linear models provides one part of the statistical theory needed on generalized additive models. With it any exponential family distribution can be used in linear modelling along with different link functions which also play important roles in the GAM. The complicated likelihood equations in GLM are solved by using the derivation chain rule that leads us to a form which allows us to use the Newton's method and iteratively solve the parameters for the model. Nonparametric regression methods, such as cubic splines and thin plate splines, on the other hand allow any form for the explanatory variables to take place. They produce so-called smooth functions that base on data and smoothness selection. Generalized linear models and nonparametric regression are then combined to generalized additive models by imitating nonparametric methods with parametric estimates.

Generalized additive models are based largely on generalized linear models. Most of the inference is same and modelling is done in the same way, although with more caution. Smoothness selection in GAM stands out as the biggest problem. It is usually solved by generalized cross-validation criterion.

A practical example from the field of psychiatry is presented. Mothers' depression symptoms and adolescents psychosocial problems are modelled by a generalized additive model to illustrate one possible usage of GAM. When modelling adolescent's externalizing problem score mother's depression symptoms are found to be uninfluential on adolescent's symptoms. Although mother's opinion on adolescent's well-being is a predictive factor.

Keywords: generalized linear model, nonparametric regression, natural cubic spline, cubic regression spline, thin plate spline, tensor product smoother, Child Behavior Checklist, Young Self Report

Contents

1 Introduction

Traditional linear regression is perhaps the most known statistical modelling method and in addition for being also simple it is still very useful in many situations. It has been extended among others to linear mixed model and generalized linear model. Though mixed models and generalized linear models work well in some special situations, e.g. longitudinal or binary data, there are yet datas that do not fit into a parametric framework. Generalized linear models are represented in the second chapter along with some basic inference on them.

Another point of view has been nonparametric regression methods which do not assume any particular form of the data. Natural cubic splines, thin plate splines and e.g. adaptive splines all work in a different way to describe the data. And that is what they mostly do. These type of methods tend to use all the data but they can be imitated with regression and dimension reduction methods. Spline type of methods work well in describing the data and also in testing linearity. That is aswering the question whether a parametric form fits well enough or should a parametric hypothesis be rejected. Nonparametric mehods are discussed in the third chapter and also the parametric estimates of those methods are represented to give a glimpse of how they work.

Hastie and Tibshirani integrated generalized linear models and nonparametric spline methods into generalized additive models (GAM) in 1986 and 1990 with their *backfitting algorithm*. In 2006 Wood released his version of GAM with some improvements and extensions on it. A generalized additive model allows parametric and nonparametric terms to be added in the same model, which gives us the possibility for more flexible modelling. Chapters four and five discus GAM and some of its inference. In the sixth chapter we present a practical example on modelling mothers' depression symptoms and adolescents' behavioral symptoms.

2 Generalized linear model

The class of generalized linear models is an extension of classical linear models. In the basic model structure (2.1) the only new part is a link function $g(.)$ that transforms the expectation of $E(Y)$ to linear space. This generalization allows us to model several kinds of explanatory variables with a linear predictor $\eta_i = \sum_{j=1}^{p} x_{ij}\beta_j$, which is exactly of the same form as in e.g. linear regression but due to the link function the interpretation of parameters may change. The model formula becomes

$$g(\mu_i) = \sum_{j=1}^{p} x_{ij}\beta_j \tag{2.1}$$

where $\mu_i = E(y_i)$. Beyond the model formula is the choise of a probability distribution of $\boldsymbol{Y}$. The GLM allows all distributions of exponential family and so-called quasi-likelihoods. The choise of a distribution also restricts the choise of possible link functions. Due to different distributions and generalization of linearity we have to define also more general estimation equations. (McCullach, P. and Nelder, J.A. 1983.)

2.1 Exponential family of distributions

A Distribution belongs to the family of exponential distributions if its probability density (or mass) function can be written in a certain form (Theorem (2.1)). Exponential family of distributions includes many of the most common distributions, e.g. normal, gamma, poisson and binomial distributions.

Theorem 2.1. *(McCullach, P. and Nelder, J.A. 1983) A distribution belongs to the family of exponential distributions if it's probability density function or probability mass function can be written as*

$$f(y;\theta,\phi) = exp\left\{(y\theta - b(\theta))/a(\phi) + c(y,\phi)\right\}, \tag{2.2}$$

where a, b and c are some functions, ϕ a scale parameter and θ 'canonical parameter'.

Example 2.1. If $Y \sim Gamma(\alpha, \frac{\alpha}{\beta})$ then it can be shown that Y has an exponential family distribution. To make the calculations easier it is useful to mark the beta parameter as a ratio of alpha- and beta-parameters.

$$\begin{aligned} f(y;\theta,\phi) &= \frac{1}{\Gamma(\alpha)}\left(\frac{\alpha}{\beta}\right)^{\alpha} y^{\alpha-1} exp\left\{-\frac{\alpha y}{\beta}\right\} \\ &= exp\left\{-\frac{\alpha y}{\beta} + log\left(\frac{1}{\Gamma(\alpha)}\left(\frac{\alpha}{\beta}\right)^{\alpha} y^{\alpha-1}\right)\right\} \\ &= exp\left\{-\frac{\alpha y}{\beta} + log\left(\frac{1}{\beta}\right)^{\alpha} + log\left(\frac{(\alpha y)^{\alpha}}{\Gamma(\alpha) y}\right)\right\} \\ &= exp\left\{-\frac{\alpha y}{\beta} + \alpha log\left(\frac{1}{\beta}\right) + log\left((\alpha y)^{\alpha}\right) - log\left((\Gamma(\alpha) y)\right)\right\} \\ &= exp\left\{\frac{-\frac{y}{\beta} + log\left(\frac{1}{\beta}\right)}{\frac{1}{\alpha}} + \alpha log\left(\alpha y\right) - log\left(y\Gamma(\alpha)\right)\right\}, \end{aligned}$$

where $\theta = -\frac{1}{\beta}$, $b(\theta) = -log\left(-\theta\right)$, $a(\phi) = \frac{1}{\alpha}$ and $c = \alpha log\left(\alpha y\right) - log\left(y\Gamma(\alpha)\right)$.

The theory of exponential family of distributions allows us to easily calculate expectation and variance of the random variable Y, which are needed when we want to solve the parameters of the linear predictor in (2.1). Solving these expecatation and variance for Y is easier though if we first know the expectation and variance of the log-likehood. Let us look at the expectation property first (Wood 2006). Given independent observations $y_1, y_2, \ldots, y_n$ from a p.d.f. $f(y,\theta)$ we have a log-likelihood

$$l(\theta) = \sum_{i=1}^{n} log\left[f\left(y_i,\theta\right)\right] = \sum_{i=1}^{n} l_i\left(\theta\right).$$

Now the expectation of the log-likelihood of a single observation with respect to $f(y,\theta_0)$, with θ_0 as a true value of θ, is

$$\begin{aligned} E_0\left(\frac{\partial l_i}{\partial \theta}\right) &= E_0\left(\frac{\partial}{\partial \theta} log\left[f\left(y_i,\theta\right)\right]\right) = \int \frac{1}{f\left(y,\theta_0\right)}\frac{\partial f}{\partial \theta} f\left(y,\theta_0\right) dy \\ &= \int \frac{\partial f}{\partial \theta} dy = \frac{\partial}{\partial \theta}\int f dy = \frac{\partial 1}{\partial \theta} = 0, \end{aligned}$$

where the third part follows from the derivative of a logaritmic function and from the expectation-integral. The expectation above is taken with respect to one observation but holds for n observations as well. Since variance can always be calculated from the well known result $\text{Var}\left(Y\right) = \left[E\left(X^2\right)\right] + \left[E\left(X\right)\right]^2$, we immediately get

$$\text{Var}_0\left(\frac{\partial l}{\partial \theta}\right) = E_0\left[\left(\frac{\partial l}{\partial \theta}\right)^2\right]. \tag{2.3}$$

The expectation in (2.3) is considered to be the *Fisher information* about θ in the data. It is useful to notice (see Wood 2006) that it can be represented as

$$E_0\left[\left(\frac{\partial l}{\partial \theta}\right)^2\right] = -E_0\left[\left(\frac{\partial^2 l}{\partial \theta^2}\right)\right]. \tag{2.4}$$

Now, to get back to the problem of solving expectation and variance of Y, for log-likelihood we have

$$l(y;\theta,\phi) = (y\theta - b(\theta))/a(\phi) + c(y,\phi) \tag{2.5}$$

and differentiations of the log-likelihood give

$$\frac{\partial l}{\partial \theta} = \frac{y - b'(\theta)}{a(\phi)} \tag{2.6}$$

and

$$\frac{\partial^2 l}{\partial \theta^2} = \frac{-b''(\theta)}{a(\phi)}. \tag{2.7}$$

Now, since we know that the expecation of the first derivative should be zero, we get by taking the expectation

$$\begin{aligned} & E\left(\frac{\partial l}{\partial \theta}\right) = E\left(\frac{y - b'(\theta)}{a(\phi)}\right) = \frac{E(y) - b'(\theta)}{a(\phi)} = 0 \\ & \Leftrightarrow E(y) = \mu = b'(\theta). \end{aligned} \tag{2.8}$$

Using the property of information from (2.4) we get the variance

$$\begin{aligned} & E\left[\left(\frac{\partial l}{\partial \theta}\right)^2\right] = -E\left[\left(\frac{\partial^2 l}{\partial \theta^2}\right)\right] \Leftrightarrow \left(\frac{y - b'(\theta)}{a(\phi)}\right)^2 = -\left(\frac{-b''(\theta)}{a(\phi)}\right) \\ & \Leftrightarrow \frac{[y - E(y)]^2}{[a(\phi)]^2} = \frac{b''(\theta)}{a(\phi)} \Leftrightarrow \frac{\mathrm{Var}(y)}{[a(\phi)]^2} = \frac{b''(\theta)}{a(\phi)} \\ & \Leftrightarrow \mathrm{Var}(y) = \frac{b''(\theta)}{a(\phi)}[a(\phi)]^2 = b''(\theta)a(\phi). \end{aligned} \tag{2.9}$$

We now have expectation and variance in a form that links us back to the theory on exponential family of distributions. So it is rather easy to solve those for any random variable that has an exponential family distribution. This comes handy when solving maximum likelihood estimates for GLM.

Example 2.2. For a random variable Y that has a gamma distribution (see example (2.1)) we have

$$\begin{aligned} & E(Y) = b'(\theta) = \frac{\partial}{\partial \theta} - log\,(-\theta) = -\frac{1}{-\theta}(-1) = -\frac{1}{\theta} = \beta \text{ and} \\ & \mathrm{Var}(Y) = b''(\theta)a(\phi) = \frac{1}{\alpha}\frac{\partial}{\partial \theta}\left(-\frac{1}{\theta}\right) = \frac{1}{\alpha}\frac{1}{\theta^2} = \frac{1}{\alpha\left(-\frac{1}{\beta}\right)^2} = \frac{\beta^2}{\alpha}. \end{aligned}$$

2.2 Maximum likelihood for GLM

We are naturally interested of the estimates in the parameters β_j in the linear predictor (see derivatives in (2.1)). It is possible to solve likelihood equations with respect to each β_j. Derivation of the log-likelihood in (2.5) are by chain rule

$$\frac{\partial l_i}{\partial \beta_j} = \frac{\partial l_i}{\partial \theta_i}\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta_j}. \tag{2.10}$$

Now, the first two partial derivatives can be solved separately to ease up the derivation.

$$\begin{aligned}\frac{\partial l_i}{\partial \theta_i} &= \frac{y_i - b'(\theta_i)}{a(\phi)} = \frac{y_i - \mu_i}{a(\phi)},\\ \frac{\partial \theta_i}{\partial \mu_i} &= \left(\frac{\partial \theta_i}{\partial \mu_i}\right)^{-1} = \left(b''(\theta)\right)^{-1} = \frac{a(\phi)}{\mathrm{Var}(y_i)}.\end{aligned}$$

By combining these obvious results into the equation (2.10) we can easily transform the derivative into a suitable form

$$\frac{\partial l_i}{\partial \beta_j} = \frac{y_i - \mu_i}{a(\phi)}\frac{a(\phi)}{\mathrm{Var}(y_i)}\left(\frac{\partial \mu_i}{\partial \beta_i}\right) = \frac{y_i - \mu_i}{\mathrm{Var}(y_i)}\left(\frac{\partial \mu_i}{\partial \beta_j}\right), \tag{2.11}$$

and this immediately gives us the likelihood maximization problem of β_j as neat sum of such equations

$$\sum_{i=1}^{n}\frac{y_i - \mu_i}{\mathrm{Var}(y_i)}\left(\frac{\partial \mu_i}{\partial \beta_j}\right) = 0,\ 1 \leq j \leq p. \tag{2.12}$$

There are several ways to continue solving the problem. First we have to take a look at the variance of y_i, $\mathrm{Var}(y_i) = b''(\theta_i)a(\phi)$, where $a(\phi)$ is often a known function of ϕ. Though it is not always known so it is more useful to define $a(\phi) = \phi/\omega$, where ω is a known constant. This makes $\mathrm{Var}(y_i) = b''(\theta_i)\phi/\omega$ and now defining variance as a function of expectation μ_i we can define funtion $V(\mu_i) = b''(\theta_i)/\omega$, so that $\mathrm{Var}(y_i) = V(\mu_i)\phi$. Since ϕ and ω are both constants we do not need to input them to the equations in (2.12). Furthermore the estimation equations can be reduced to an equivalent non-linear least squares problem (see Wood 2006) when μ_i is considered as a function of β_j's

$$\sum_{i=1}^{n}\frac{(y_i - \mu_i)^2}{V(\mu_i)}. \tag{2.13}$$

In matrix form we have

$$S = \left\|\sqrt{\boldsymbol{V^{-1}}}\,(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{\beta}))\right\|^{\mathbf{2}}, \tag{2.14}$$

where $\boldsymbol{V}$ is a diagonal matrix with $V_{ii} = V(\mu_i)$ and $\boldsymbol{\mu(\beta}_{)}$ an estimated $\boldsymbol{\mu}$ based on $\boldsymbol{\beta}$. What comes to this minimizing problem, from (2.13) it's easy to suggest that any dependence between expectation and variance could be placed in. If one defines own functions for link and variance then (2.13) is a non-linear least squares for so-called quasi-likelihood. Not all choises produce converging minimization though (see Wood 2006). Since we are now dealing with a non-linear least squares it is convenient to use a first order Taylor expansion of $\boldsymbol{\mu}$ around $\widehat{\boldsymbol{\beta}_{[k]}}$. With $\boldsymbol{V}$ also changed to its current estimate we have

$$S = \left\| \sqrt{\boldsymbol{V}_{[k]}^{-1}} \left(\boldsymbol{y} - \boldsymbol{\mu}^{[k]} - \boldsymbol{J}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{[k]}) \right) \right\|^{\mathbf{2}},$$

where $\boldsymbol{J}$ is the Jacobian matrix i.e. the matrix of the first derivatives of μ_i w.r.t. β_j for all i, j. With a few simple notions (see Wood 2006) this can be turned into

$$S = \left\| \sqrt{\boldsymbol{W}^{[k]}} \left(\boldsymbol{z}^{[k]} - \boldsymbol{X\beta} \right) \right\|^{\mathbf{2}}, \tag{2.15}$$

where $z_i^{[k]} = g'(\mu_i^{[k]}) \left(y_i - \mu_i^{[k]} \right) + \eta_i^{[k]}$ is the pseudodata that takes the least squares back to linear space and $\boldsymbol{W}^{[k]}$ is the diagonal weight matrix with elements

$$W_{ii}^{[k]} = \frac{1}{V(\mu_i^{[k]}) g'(\mu_i^{[k]})^2}.$$

Finally, with some initial values of $\boldsymbol{\mu}^{[k]}$ and $\boldsymbol{\eta}^{[k]}$ we can calculate a new pseudodata $\boldsymbol{z}^{[k]}$ and weights $W^{[k]}$ and then minimize (2.15) to get $\boldsymbol{\beta}^{[k+1]}$, $\boldsymbol{\eta}^{[k+1]}$ and $\boldsymbol{\mu}^{[k+1]}$. This is repeated until there is no significant change in the parameter estimates. The algorithm sketched here is Newton-Raphson based Iteratively Re-weighted Least Squares (IRLS) but also Fisher scoring method can be used to obtain the estimates for $\boldsymbol{\beta}$. (Wood 2006, McCullach, P. and Nelder, J.A. 1983.)

As seen before, the choise of distribution and link function have an effect on the estimation equations. Actually, each distribution has a canonical link which means that $g_c(\mu_i) = \theta_i = \boldsymbol{X_i\beta}$. When modelling with a canonical link the estimation equations reduce to a simpler form since in (2.10) $\frac{\partial \theta_i}{\partial \beta_j} = X_{ij}$ and

$$\frac{\partial l_i}{\partial \beta_j} = \frac{\partial l_i}{\partial \theta_i} \frac{\partial \theta_i}{\partial \beta_j} = \frac{y_i - b'(\theta_i)}{a(\phi)} X_{ij} = \frac{y_i - \mu_i}{a(\phi)} X_{ij}.$$

Canonical link is thus a natural choise to make. E.g. for normal distribution identity link is the canonical link and it leads to standard linear regression, and for gamma distribution inverse link is the canonical link function. (Wood 2006.)

2.3 Inference in GLM

Inference is, of course, the main thing in statistical modelling but only the most important results are represented here. As in the standard regression both model comparison and residual checking rise up when choosing an adequate model. Model comparison needs to be done with a little more general methods than in linear regression and also standardising residuals require some adjustment. Checking whether the fitted and observed values match close enough is always another side of the coin when ensuring that the model is anywhere near right.

2.3.1 Deviance and AIC

Widely used deviance is based on log-likelihoods of the model in inspection and saturated model and ω_i is the weight of the ith observation. Saturated model means a full model that has n parameters and so its log-likelihood has the absolute highest value. Basically deviance is a sum of the differences of log-likelihoods for all the data points

$$\begin{aligned} D &= 2\left[l(\hat{\boldsymbol{\beta}}_{\boldsymbol{max}}) - l(\hat{\boldsymbol{\beta}})\right]\phi \\ &= \sum_{i=1}^{n} 2\omega_i \left[y_i(\hat{\theta}_{max} - \hat{\theta}) - b(\hat{\theta}_{max}) + b(\hat{\theta})\right] = \sum_{i=1}^{n} d_i, \end{aligned}$$

where $l(\hat{\boldsymbol{\beta}})$ is the log-likelihood of the model in inspection. Scaling the deviance is needed when the scale parameter ϕ is not known to be 1, in practice whenever $\boldsymbol{Y}$ is not binomial or Poisson distributed. Scaled deviance is $D^* = D/\phi$ which can be only approximated to have a chi-square distribution with $n-p$ degrees of freedom i.e. $D^* \sim \chi^2_{n-p}$. Now the likelihood ratio test can be performed to compare null model and an alternative model. When $D_0^* \sim \chi^2_{n-p_0}$ and $D_1^* \sim \chi^2_{n-p_1}$ we have

$$D_0^* - D_1^* \sim \chi^2_{n-p_0} - \chi^2_{n-p_1} \sim \chi^2_{p_1-p_0} \tag{2.16}$$

under H_0. Not knowing the scale parameter ϕ doesn't matter since we may turn this test into an F-test. We know that the difference of the deviances and the deviance of the alternative model both have chi-square distributions, that is the ratio of these two is F-distributed. Recalling the scale parameters we have

$$F = \frac{(D_0/\phi - D_1/\phi)/(p_1 - p_0)}{(D_1/\phi)/(n - p_1)} = \frac{(D_0 - D_1)/(p_1 - p_0)}{D_1/(n - p_1)} \sim F_{p_1-p_0,n-p_1}. \tag{2.17}$$

Though deviance is good for choosing parameters into the model its distributional assumption is wrong since it relies on the number of parameters staying fixed as the number of observations tends to infinity. But the saturated model has as many parameters as observations and thus it is not totally right to compare a model to the saturated model with deviance. Still asymptotic results may give good enough results and for the Normal model it is exact. (Wood 2006, McCullach, P. and Nelder, J.A. 1983.)

Another approach in choosing a model is Akaike's information criterion. Wood (2006) justifies this by showing a straight equivalence between minimizing AIC and maximizing log-likelihood. AIC for a GLM is simply

$$AIC = 2\left[-l(\hat{\boldsymbol{\beta}}) + p\right], \tag{2.18}$$

when ϕ is known. Otherwise, if ϕ is estimated then p is replaced by $p+1$ because of an extra parameter.

2.3.2 Residuals

Calculating deviance involves calculating separate quantities d_i which as a sum form the actual deviance. To get the actual deviance residuals we just have to add the right sign for each term $\sqrt{d_i}$. The deviance residual becomes

$$\epsilon_i^d = sign(y_i - \hat{\mu}_i)\sqrt{d_i} \tag{2.19}$$

Since deviance has a chi-square distribution with $n - p$ degrees of freedom the single datums d_i can be assumed to have approximately a chi-square distribution with 1 degree of freedom (when we assume all parameters known, i.e. $p = 0$) and thus deviance residuals have a standard normal distribution $\epsilon_i^d \sim N(0, 1)$. And thus checking residuals is an analogue to regression models residual check. Another similar analogue is the Pearson's statistic and the Pearson residuals which inherit from poisson distribution situation. Pearson's statistic is simply

$$X^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}, \tag{2.20}$$

which is exactly the same as the residuals in (2.13) with the μ_i's replaced by their maximum likelihood estimates. Square roots of the single datums of Pearson's statistic are the Pearson residuals, which are standardized by their variances.

Another way to standardize residuals is to take into account how big effect each observation has on the expected value. Firstly, the projection matrix for GLM is $\boldsymbol{H} = \sqrt{\boldsymbol{W}}\boldsymbol{X}(\boldsymbol{X'WX})^{-1}\boldsymbol{X'}\sqrt{\boldsymbol{W}}$. Since $\boldsymbol{H}$ maps a certain proportion (< 1) of each observation to each expected value and the rowsums of a projection matrix are all 1 it's a natural choise to use $\boldsymbol{h_{ii}}$'s in standardizing residuals. Studentized residuals make use of this property. Dividing residuals by $\sqrt{1-h_i}$ (where h_i stands for $\boldsymbol{h_{ii}}$) gives studentized residuals from standard, pearson and deviance residuals. Furthermore deletion residuals are calculated by omitting the ith observation from calculation of expectation and variance and then calculating the ith residual. Now, as an example, standard residual as studentized deletion residual can be calculated as

$$r_i^* = \frac{y_i - \hat{\mu}_{(i)}}{s_{(i)}\sqrt{1 + h_{(i)}}} = \frac{y_i - \hat{\mu}_i}{s_{(i)}\sqrt{1 - h_i}}. \tag{2.21}$$

Since the model comparison by deviance and F-test is not enough it is always necessary to take a good look at the residuals. It is recommended to plot standardized residuals against the predicted values and see if there is some systematic departure. It is often the case that variance tends to increase as the predicted values increase. In that case some other variance function or alternatively other link function should be used. When modelling a binomial or poisson model it might be helpful to just let the dispersion parameter vary. If the model is constructed carefully from the beginning and the structure ca not be changed additional covariates might be the right way to try. In a case that the residuals *vs* linear predictor or residuals *vs* a single covariate plots show a linear trend or some curvature it might be an indication of a wrong choise for link function or of a power term of wrong degree.

In addition to systematic departures isolated data points may exist and affect on the statistics of the model, either in a good or in a bad way. Points that are that greatly change the parameter estimates are the worst case scenario but also uneffective points that make a big difference on the goodness-of-fit statistics are unsatisfactory. With a large data that has some small group of data points different from the rest of the data one could make a new dummy variable to isolate them from the big picture. (Wood 2006, McCullach, P. and Nelder, J.A. 1983.)

3 Generalized additive model and smoothing theory

W e start this section with a short introduction of general structure of a GAM i.e. a generalized additive molel. The basic theory of GAM involves also some discussion on smoothing methods and smoothness selection. Reader is assumed to have the basic knowledge on generalized linear models, introduced earlier, and nonparametric regression.

3.1 Introducing generalized additive model

Generalized additive models were firstly proposed by Hastie and Tibshirani (1986 and 1990). The basic structure of a GAM (3.1) can be represented as generalized linear model involving smooth functions and smooth interactions (Wood 2006, p. 121).

$$g(\mu_i) = \boldsymbol{x}_i^* \boldsymbol{\theta} + f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}, x_{4i}) + \ldots \tag{3.1}$$

where $\mu_i = E(Y_i)$, $g(\mu_i)$ is some link function and the response variable Y_i follows some esponential family distribution. And $\boldsymbol{x}_i^*$ is a row of the model matrix for usual parametric part of the model, $\boldsymbol{\theta}$ the corresponding parameter vector and f_j are smooth functions of some covariates x_k.

As it stands out a generalized additive model can be derived from a generalized linear model by relaxing the linearity constraint of a linear predictor. That is we do not assume a parametric form for the linear predictor. However, to efficiently estimate a GAM, it turns out that we have to use a parametric form to approximate the nonparametric functions (see section 3.2.2).

3.2 Applicable smoothing methods

The smooth functions f_j in a generalized additive model (3.1) are estimated by nonparametric regression methods such as cubic regression splines or, in a multivariate case, thin plate regression splines or tensor product smoothers. There are also other smoothing methods available in R-package mgcv (see Wood 2006) but not all the methods are represented here. In order to discuss these methods it is useful to take a look at penalized regression first (following Green and Silverman 1994).

Penalized regression problem arises from two basic principles of statistical curve fitting. Firstly, we want an estimated curve g to fit the given data well. But we also need to simplify the result so that the curve is not following any rapid fluctuations i.e. modelling noise. There is an extensive amount of literature on penalized regression. A widely used solution is to measure the curve "wiggliness" by its integrated squared second derivative, assuming that g is twice-differentiable. Green and Silverman (1994) justify the second derivative by a reasonable fact that we do not wish wiggliness to be affected by a constant or a linear function. This approach is often generalized to m times differentiable functions (see e.g. Wahba 1990) but in the framework of GAMs we only need second order derivatives. So, let us consider a penalized regression problem of the form

$$S(g) = \sum \{Y_i - g(t_i)\}^2 + \alpha \int_a^b \{g''(x)\}^2 \, dx, \tag{3.2}$$

where α is called the smoothing parameter. It determines how smooth the result should be, although there is no easy solution how to choose α. In R-package called `mgcv` (multiple generalized cross validation) smoothing parameter selection is solved by generalized cross-validation i.e. GCV (see section 4.1) or UBRE score which minimize the error when predicting new data (Wood 2006). In order to get a more mathematical aspect on the issue we shall first consider interpolating the data i.e. set $\sum \{Y_i - g(t_i)\}^2 = 0$.

3.2.1 Natural cubic spline

When interpolating the data in (3.2) cubic splines arise as a mathematical solution to minimize the penalizing term $\int_a^b \{g''(x)\}^2 \, dx$. We shall now discover the mathematical conditions of cubic splines and natural cubic splines.

Natural cubic spline. *(Green and Silverman 1994) Given ordered real numbers $t_1, \ldots, t_n \in [a, b]$ satisfying $a < t_1 < t_2 < \ldots < t_n < b$, a function g is a cubic spline if*

1. *g is cubic polynomial on each interval $(a, t_1), (t_1, t_2), \ldots, (t_n, b)$ and*
2. *g, g' and g'' are continuous at each point t_i. And g is said to be a natural cubic spline (NCS) if also*
3. *$g''(a) = g''(b) = g'''(a) = g'''(b) = 0$*

The first two conditions of a natural cubic spline define a function space $S[a, b]$ that contains all functions g that have two continuous derivatives. These conditions imply that g'' is integrable so that $\int_a^x g''(x)dt = g'(x) - g'(a)$, which means that g belongs to Sobolevs space $S_2[a, b]$. Now we can show that the NCS interpolant minimizes the penalizing term in (3.2) over all functions in $S_2[a, b]$.

Theorem 3.1. *(Green and Silverman 1994) Suppose $n \geq 2$ and that g is the NCS with $g(t_i) = z_i$ for $i = 1, \ldots, n$ satisfying $a < t_1 < t_2 < \ldots < t_n < b$. Given any interpolant $\tilde{g} \in S_2\,[a, b]$ (i.e. $\widetilde{g(t_i)} = z_i$ for $i = 1, \ldots, n$) it holds that $\int_a^b \{g''(x)\}^2\, dx \leq \int_a^b \{\tilde{g}''(x)\}^2\, dx$.*

Proof. Defining $h = \tilde{g} - g$, integrating $\tilde{g}'' = g'' + h''$ gives

$$\begin{aligned}
\int\limits_a^b \tilde{g}''^2 &= \int\limits_a^b (g'' + h'') \\
&= \int\limits_a^b g''^2 + 2\int\limits_a^b g''h'' + \int\limits_a^b h''^2 \\
&= \int\limits_a^b g''^2 + \int\limits_a^b h''^2,
\end{aligned}$$

since $g''(a) = g''(b) = 0$ and $h(t_i) = \tilde{g}(t_i) - g(t_i) = 0$ and thus

$$\begin{aligned}
\int\limits_a^b g''(t)h''(t)dt &= \mathop{\Big/}\limits_a^b g''(t)h'(t) - \int\limits_a^b g'''(t)h'(t)dt \\
&= -\sum_{j=1}^{n-1} g'''(t_j^+) \int\limits_{t_j}^{t_{j+1}} h'(t)dt \\
&= -\sum_{j=1}^{n-1} g'''(t_j^+) \left\{h(t_{j+1}) - h(t_j)\right\} = 0.
\end{aligned}$$

Furthermore

$$\int\limits_a^b g''^2 + \int\limits_a^b h''^2 \geq \int\limits_a^b g''^2,$$

which equals only if h is linear on $[a, b]$. So h needs to be identically zero i.e. $\tilde{g}$ is g, since $h(t_i) = 0$ for all i. □

Suppose we know that g is a NCS with knots $t_1 < t_2 < \ldots < t_n$. Knowing the values $g_i = g(t_i)$ and second derivatives $\gamma = g''(t_i)$ for $i = 1, \ldots, n$, we get a very neat and useful representation for the NCS (Green and Silverman 1994). We define $\boldsymbol{g} = (g_1, \ldots, g_n)'$ and $\boldsymbol{\gamma} = (\gamma_2, \ldots, \gamma_{n-1})'$, since $\gamma_1 = \gamma_n = 0$ by definition. Now, vectors $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ can be used to specify g completely but also to define if vectors $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ really specify a natural cubic spline. Define $h_i = t_{i+1} - t_i$ for $i = 1, \ldots, n-1$ and $n \times (n-2)$, now matrix $\boldsymbol{Q}$ for $i = 1, \ldots, n$ and $j = 2, \ldots, n-1$ has entries

$$q_{j-1,j} = h_{j-1}^{-1}, \qquad q_{jj} = -h_{j-1}^{-1} - h_j^{-1} \qquad q_{j+1,j} = h_j^{-1}$$

with other entries 0. Now define $(n-2) \times (n-2)$ tridiagonal matrix $\boldsymbol{R}$ with entries r_{ij} for $i = j = 2, \ldots, n-1$

$$r_{ii} = \frac{1}{3}(h_{i-1} - hi), \qquad r_{i,i+1} = r_{i+1,i} = \frac{1}{6}h(i).$$

With the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ we can define an important condition.

Theorem 3.2. *The vectors $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ can represent a natural cubic spline iff*

$$\boldsymbol{Q}'\boldsymbol{g} = \boldsymbol{R}\boldsymbol{\gamma} \tag{3.3}$$

and if true then the penalizing term can be stated as

$$\int_a^b g''^2 dt = \boldsymbol{\gamma}\boldsymbol{R}\boldsymbol{\gamma} = \boldsymbol{g}'\boldsymbol{K}\boldsymbol{g}, \text{ where } \boldsymbol{K} = \boldsymbol{Q}\boldsymbol{R}^{-1}\boldsymbol{Q}'. \tag{3.4}$$

Green and Silverman (1994) give a thorough proof for statements in Theorem (3.2). From this representation it is obvious that after all we do not actually need the second derivatives, as given, in order to estimate a natural cubic spline. To get back to our original smoothing problem, consider $\hat{g}$ as a minimizer to equation (3.2). Now it is easy to show that $\hat{g}$ immediately is a natural cubic spline even in the case that we are not interpolating the data.

Theorem 3.3. *Given g that is not an NCS with knots at $t_1, \ldots, t_n$ there is always a natural cubic spline which minimizes the penalized least squares in* (3.2).

Proof. If $\hat{g}$ is natural cubic spline interpolant for values $g(t_i)$ then

1. $\sum \{Y_i - \hat{g}(t_i)\}^2 = \sum \{Y_i - g(t_i)\}^2$, since $\hat{g}$ is an interpolant and
2. $\int_a^b \{\hat{g}''(x)\}^2 dx \leq \int_a^b \{g''(x)\}^2 dx$ (Theorem (3.1)).

It follows from the definition of NCS that a natural cubic spline always attains smaller value in penalized least squares. □

From equation (3.2) and Theorem (3.2) it is easy to obtain a formal solution for curve g whether estimating a natural cubic spline interpolant or a cubic smoothing spline. However, it would be rather inconvenient to use this kind of approach in GAM due to the fact that a smoothing spline uses n degrees of freedom. So, we have to take another step to obtain a more practical way of estimation.

3.2.2 Cubic regression spline

There are several ways to represent a natural cubic spline as a sum of basis functions that evaluate the spline in data points (Wood 2006). An easy solution would be to use a third degree truncated power basis (Wu and Zhang 2006) whereas a B-spline basis (de Boor 1978, Eubank 1988) is a more popular choice. For more mathematical expression of splines one should see Wahba (1990) for reproducing kernel Hilbert spaces that can be used in splines of any degree.

Another practical way to turn spline smoothing into regression is to use basis functions that yeild from interpolating the whole spline. For interpolating purpose Green and Silverman (1994) state a way to represent a natural cubic spline at any point $t \in [t_1, t_n]$. Given vectors $\boldsymbol{g}$ and $\boldsymbol{\gamma}$ and some sequential knots t_L and t_R, with corresponding values g_L, g_R and $h = t_R - t_L$, we have for $t \in [t_L, t_R]$

$$\begin{aligned} g(t) =& \frac{(t-t_L)g_R + (t_R - t)g_L}{h} \\ &- \frac{1}{6}(t-t_L)(t_R-t)\left\{\left(1+\frac{t-t_L}{h}\right)\gamma_R + \left(1+\frac{t_R-t}{h}\right)\gamma_L\right\}. \end{aligned} \tag{3.5}$$

Since $t_R - t = (t_R - t_L) - (t - t_L) = h - (t - t_L)$ and respectively $t - t_L = h - (t_R - t)$, we can reorder the latter part as

$$\begin{aligned} &- \frac{1}{6}(t-t_L)(t_R-t)\left\{\left(1+\frac{t-t_L}{h}\right)\gamma_R + \left(1+\frac{t_R-t}{h}\right)\gamma_L\right\} \\ =& - \frac{1}{6}(t-t_L)(h-(t-t_L)\left(1+\frac{t-t_L}{h}\right)\gamma_R - \dots \\ =& - \frac{1}{6}(h(t-t_L)-(t-t_L)^2)\left(1+\frac{t-t_L}{h}\right)\gamma_R - \dots \\ =& - \frac{1}{6}\left(h(t-t_L) + h\frac{(t-t_L)^2}{h} - (t-t_L)^2 - \frac{(t-t_L)^3}{h}\right)\gamma_R - \dots \\ =& \frac{1}{6}\left(\frac{(t-t_L)^3}{h} - h(t-t_L)\right)\gamma_R + \frac{1}{6}\left(\frac{(t_R-t)^3}{h} - h(t_R-t)\right)\gamma_L. \end{aligned}$$

Now it is easy to obtain basis functions similar to what Wood (2006, p. 149-150) represents. These so-called cardinal basis functions are also used in `mgcv`. So, now the basis functions that can fully represent a natural cubic spline can be derived as

$$\begin{aligned} a_j^-(t) = \frac{(t_R - t)}{h_j},\ c_j^-(t) = \frac{1}{6}\left[\frac{(t_R-t)^3}{h_j} - h_j(t_R - t)\right], \\ a_j^+(t) = \frac{(t - t_L)}{h_j} \text{ and } c_j^+(t) = \frac{1}{6}\left[\frac{(t-t_L)^3}{h_j} - h_j(t - t_L)\right]. \end{aligned}$$

Now define $g(t) \in [t_L, t_R]$ by (3.5) as

$$g(t) = a_j^-(t)g_R + a_j^+(t)g_L + c_j^-(t)\gamma_R + c_j^+(t)\gamma_L. \tag{3.6}$$

It is easy to see that these basis functions are continuous. By taking left-sided and right-sided derivatives from (3.6) and matching those we get exactly the condition (3.3). That is (3.6) still represents a natural cubic spline. Though with basis functions we are actually fitting a cubic regression spline and dealing with a regression problem, which makes it easier to compute and interpret (Wood 2006). By using conventional regression method in estimation we evaluate the fitted function in the data points by basis functions given above, and regress those values $g(t)$ on some explanatory variable t.

From (3.3) we see that $\boldsymbol{\gamma} = \boldsymbol{R^{-1}Qg}$ for $j = 2, \ldots, k-1$. Substituting this, with zeroes added to both ends (naturality constraint), into (3.6) $g(t)$ is fully defined by just vector $\boldsymbol{g}$ as

(3.7)

$$g(t) = a_j^-(t)g_R + a_j^+(t)g_L + c_j^-(t)\left[\boldsymbol{R^{-1}Q}\right]_{\boldsymbol{j}}\boldsymbol{g} + c_j^+(t)\left[\boldsymbol{R^{-1}Q}\right]_{\boldsymbol{j+1}}\boldsymbol{g} \equiv \sum_{i=1}^{k} b_i(t)g_i$$

Wood (2006) also shows that now the penalizing term can be written in the same form as in (3.4). In order to fit a cubic regression spline we have to choose the locations of the knots t_i. Wu and Zhang (2006) present three widely used methods. One can locate knots equally either by range or sample quantiles. Another way is to go trough all the data points and use some of various iterative methods to choose the best locations. In `mgcv` knots are located equally by sample range as a default.

3.2.3 Thin plate splines

Generalising natural cubic spline by using a multidimensional Laplacian penalty yiedls a thin plate spline (Hastie and Tibshirani 1990, originally Duchon 1977). In two dimensional case penalty is of the form

$$J(g) = \int\int \left\{ \left(\frac{\partial^2 f}{\partial x_1^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x_1 \partial x_2}\right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2}\right)^2 \right\} dx_1 dx_2. \tag{3.8}$$

Green and Silverman (1994) give a useful definition of a natural cubic spline, which can be used to get an analogue expression for a thin plate spline. We can express a natural cubic spline as

$$g(t) = a_1 + a_2 t + \frac{1}{12}\sum_{i=1}^{n} \delta_i \left|t - t_i\right|^3, \tag{3.9}$$

constrained by

$$\sum_{i=1}^{n} \delta_i = \sum_{i=1}^{n} \delta_i t_i = 0, \tag{3.10}$$

where $\delta_i = g'''(t_i^+) - g'''(t_i^-)$ is the increment in the third derivative. By a fact that the second derivative is of course linear on each interval $[t_i, t_{i+1}]$ we get slope

$$
\begin{aligned}
g''(t) &= \frac{(t-t_i)\gamma_{i+1} + (t_{i+1}-t)\gamma_i}{h_i}, \text{ and} \\
g'''(t) &= \frac{\gamma_{i+1}-\gamma_i}{h_i}, \text{ which implies} \\
\delta_i &= \frac{\gamma_{i+1}-\gamma_i}{h_i} - \frac{\gamma_i-\gamma_{i-1}}{h_{i-1}} = [\boldsymbol{Q\gamma}]_{\boldsymbol{i}}\,. \qquad (3.11)
\end{aligned}
$$

Constraints (3.10) imply that g'' and g''' are zero outside $[t_1, t_n]$. Now, define a $2 \times n$ matrix $\boldsymbol{T}$ as

$$\boldsymbol{T} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ t_1 & t_2 & \dots & t_n \end{pmatrix}$$

and matrix $\boldsymbol{E}$ with values $E_{ij} = \frac{1}{12}\left|t_i - t_j\right|^3$. With constants $a_1, a_2, \delta_1, \delta_2, \dots, \delta_n$ in vectors $\boldsymbol{a}$ and $\boldsymbol{\delta}$, a natural cubic spline can now be written in form

$$\boldsymbol{g} = \boldsymbol{E\delta} + \boldsymbol{T'a}. \qquad (3.12)$$

With the form (3.11) the penalizing term can be calculated by $\boldsymbol{\delta' E \delta}$ and so the minimizing problem becomes

$$S(\boldsymbol{g}) = \boldsymbol{(y - E\delta - T'a)'(y - E\delta - T'a) + \alpha\delta' E\delta} \qquad (3.13)$$

Thin plate spline, in two dimensions with coordinates $\boldsymbol{t} = (x, y)$, is formed by defining

$$
\begin{gathered}
\phi_1(\boldsymbol{t}) = 1,\ \phi_2(\boldsymbol{t}) = x,\ \phi_3(\boldsymbol{t}) = y,\ \eta(r) = \frac{1}{16\pi} r^2 log r^2, \\
\boldsymbol{T_{jk}} = \phi_j(\boldsymbol{t_k}) \text{ and } E_{ij} = \eta(\|\boldsymbol{t_i} - \boldsymbol{t_j}\|).
\end{gathered}
$$

with $\eta(0) = 0$.

Thin plate spline. *(Green and Silverman 1994) A function g is a thin plate spline iff it can be expressed as*

$$g(\boldsymbol{t}) = \sum_{i=1}^{n} \delta_i \eta(\|\boldsymbol{t_i} - \boldsymbol{t_j}\|) + \sum_{j=1}^{3} a_j \phi_j(\boldsymbol{t}). \qquad (3.14)$$

And if furthermore $\boldsymbol{T\delta} = \boldsymbol{0}$ then g is a natural thin plate spline. The functions $\eta(r)$ are Green's functions for Laplacian penalty (Wahba 1990). Wahba (1990) and Wood (2003, 2006) also give the general forms for $\eta(r)$ in higher dimensions and in higher order derivatives. With these definitions thin plate splines can be presented exactly in the same way as natural cubic splines in (3.13). Also thin plate splines need to be reduced if we want to use them in modelling.

3.2.4 Thin plate regression spline

A simple way to obtain a reasonable reduction in smoothing with thin plate splines would be to use knot-based approximation (Wood 2006). Choosing k knots t_i^* equation (3.14) turns into a regression

$$g(\boldsymbol{t}) = \sum_{i=1}^{n} \delta_i \eta(\left\| \boldsymbol{t_i} - \boldsymbol{t_j^*} \right\|) + \sum_{j=1}^{3} a_j \phi_j(\boldsymbol{t}). \tag{3.15}$$

Now with coefficents combined as $\boldsymbol{\beta' = (\delta', a')}$ defining $n \times (k + M)$ matrix $\boldsymbol{X}$ with

$$X_{ij} = \eta(\left\| \boldsymbol{t_i} - \boldsymbol{t_j^*} \right\|),\ j = 1, \ldots, k$$
$$X_{ij} = \phi_{j-k}(\boldsymbol{t_i}),\ j = k+1, \ldots, k+M$$

we can transform thin plate spline estimation into a regression problem

$$\boldsymbol{\|y - X\beta\|^2 + \alpha\beta' S\beta}, \tag{3.16}$$

where S is a $(k+M)\times(k+M)$ matrix with $k\times k$ block in upper left corner where $S_{ij} = \eta(\left\| \boldsymbol{t_i^*} - \boldsymbol{t_j^*} \right\|)$, and elsewhere $S_{ij} = 0$ i.e. functions ϕ_j are not penalized. This approach to thin plate regression splines works fine in one dimension but in higher dimensios it is more difficult to choose the knot locations wisely (Wood 2006). Wood (2003, 2006) has shown a more natural way of forming an approximation of a thin plate spline. Basically the idea is to truncate the space of the $\boldsymbol{\delta}$ components and again leave $\boldsymbol{a}$ unpenalized.

1. Form an eigen-decomposition $\boldsymbol{E = UDU'}$ and rearrange eigenvalues $D_{i,i}$ in increasing order and eigenvectors in matrix $\boldsymbol{U}$ respectively.
2. Plug the first k eigenvalues and -vectors in matrices $\boldsymbol{D_k}$ and $\boldsymbol{U_k}$ i.e. $\boldsymbol{E_k = U_k D_k U_k'}$.
3. Restrict $\boldsymbol{\delta = U_k \delta_k}$ (i.e. $\boldsymbol{\delta_k = U_k^{-1} \delta_k}$) then $\boldsymbol{E_k \delta = U_k D_k U_k U_k^{-1} \delta_k = U_k D_k \delta_k}$ and $\boldsymbol{\delta' E_k \delta = \delta_k' U_k^{-1} U_k D_k U_k U_k^{-1} \delta_k = \delta_k' D_k \delta_k}$. The naturality constraints now become $\boldsymbol{T\delta = TU_k \delta_k = 0}$.

Now the fitting objective becomes

$$S(\boldsymbol{g}) = \boldsymbol{\|y - U_k D_k \delta_k - T'a\|^2 + \alpha \delta_k' D_k \delta_k} \text{ constrained by } \boldsymbol{TU_k \delta_k = 0}$$

One more trick that has to be performed is to find such matrix $\boldsymbol{Z_k}$ that $\boldsymbol{TU_k Z_k = 0}$. This can be done using QR-decomposition (Wood 2006, p. 157). Finally we have vector $\boldsymbol{\delta_k = Z_k \tilde{\delta}}$ that can be used to minimise a thin plate regression spline

$$S(\boldsymbol{g}) = \boldsymbol{\|y - U_k D_k \delta_k - T'a\|^2 + \alpha \tilde{\delta}' Z_k' D_k Z_k \tilde{\delta}}.$$

We can furthermore define $\boldsymbol{\beta' = (\tilde{\delta}', a')}$, $\boldsymbol{X = (U_k D_k \delta_k, T')}$ and $\boldsymbol{S}$ with $\boldsymbol{Z_k' D_k Z_k}$ in upper left and $\boldsymbol{0}$ elsewhere. These definitions yield exactly the same form as in (3.16).

3.2.5 Tensor product smoothers

Though thin plate splines are mathematically rather ideal smoothers they happen to be isotropic i.e. wigglines penalty is treated in the same way in each direction. Which is actually good if the predictors are of the same type and on the same scale, e.g. co-ordinates, but not when measurements are not of the same type at all. Tensor product smoothers provide a better way of solving this kind of problem. (Wood 2006.)

Tensor product smoothing bases on de Boor's (1978) idea of multiplying one linear function space U with another's say V's each object, i.e. forming a tensor product $U \times V$ of two function spaces. If U is defined on $X \in \mathbb{R}$ and V on $Y \in \mathbb{R}$ then for $(x, y) \in X \times Y$ tensor product is $w(x, y) \equiv u(x)v(y)$, where $u \in U$ and $v \in Y$. This idea can be generalized for any finite number of linear function spaces. Now assume that we have for example (see Wood 2006) three covariates x, y and z with corresponding low rank bases of the form

$$f_x(x) = \sum_{i=1}^{I} \alpha_i a_i(x),\ f_z(z) = \sum_{l=1}^{L} \delta_l d_l(z),\ f_v(v) = \sum_{k=1}^{K} \beta_k b_k(v).$$

Combining these three bases by using tensor product approach we get a three-dimensional basis function

$$f_{xzv}(x, z, v) = \sum_{i=1}^{I} \sum_{l=1}^{L} \sum_{k=1}^{K} \beta_{ilk} a_i(x) d_l(z) b_k(v),$$

where β_{ilk} is a combined coefficent for the product. If the separate bases are in matrices $\boldsymbol{X_x, X_z}$ and $\boldsymbol{X_v}$ then a model matrix for a tensor product smoother is defined with rows like $\boldsymbol{X_i = X_{xi} \otimes X_{zi} \otimes X_{vi}}$. To measure function wigglines we assume the marginal penalties can be expressed in quadratic form

$$J_x(f_x) = \boldsymbol{\alpha' S_x \alpha},$$

which is the case for smoothers presented in earlier sections. Since it is always possible to express marginal functions as

$$f_x(x) = \sum_{i=1}^{I} \alpha_i(z, v) a_i(x).$$

we can find a reasonable re-parameterization. The approach that is used in `mgcv`-package is to simply evaluate the functions $a_i(x)$ in knots x_j^* and use $\boldsymbol{\alpha' = \Gamma \alpha}$, where $\Gamma_{ij} = a_i(x_j^*)$, to re-parameterize $\boldsymbol{X_x' = X_x \Gamma^{-1}}$ and $\boldsymbol{S_x' = \Gamma S_x \Gamma^{-1}}$. With this parameterization the marginal penalty of x becomes

$$J_x^*(f_{xzv}) = \boldsymbol{\beta' \tilde{S} \beta}, \text{ where } \boldsymbol{\tilde{S} = S_x' \otimes I_L \otimes I_K}$$

and the whole penalty is a sum of such marginal penalties

$$J^*(f_{xzv}) = \lambda_x J_x^*(f_{xzv}) + \lambda_z J_z^*(f_{xzv}) + \lambda_v J_v^*(f_{xzv})$$

with corresponding smoothing parameters. (Wood 2006.)

4 Model estimation

After introducing the nonparametric regression methods we need to plug them in to the GLM context to make it a GAM. As seen before each nonparametric function is possible to estimate with the aid of parametric basis functions. By simply adding these basis functions to the model matrix and respectively adding new parameter for each basis function we have a similar setting as in the GLM. When solving a GAM the only addition to GLM in the non-linear least squares is the penalty term. Though adding it does not make the estimation any harder but it yields a new problem. Now the estimation problem (as before in equation (2.15)) becomes

$$PLS = \left\| \sqrt{\boldsymbol{W}^{[k]}} \left(\boldsymbol{z}^{[k]} - \boldsymbol{X\beta} \right) \right\|^{\boldsymbol{2}} + \boldsymbol{\beta' S \beta}, \tag{4.1}$$

where $\boldsymbol{S} = \sum_j \lambda_j \boldsymbol{S_j}$ and all the smoother matrices $\boldsymbol{S_j}$ are non-zero only where they have corresponding parameters in the parameter vector $\boldsymbol{\beta}$. In addition, all the smoothers are centered by linear constraints which force each smoothers sum to zero. This is a way to ensure that model is identifiable. It might seem like a drawback but it does not prevent us from seeing the relation between the linear predictor and covariate variable. When estimating $\boldsymbol{\beta}$ the penalty term makes it substantially harder since we would of course want to get the best ratio of accuracy and simplicity.

4.1 Smoothing parameter selection

The smoothing parameters λ_j regulate the smoothness of each smoother term. If we knew the best, or adequate, values for λ_j we could just estimate $\boldsymbol{\beta}$ straight from (4.1). For a model with one smooth function it might be tolerable to find an adequate value by trying but with more smooth functions it would come significantly harder. To visualize the effect of smoothing parameters consider the two smoother GAM in Figure 4.1. The two in the upper row are smoothed by automatic smoother selection (GCV) and the two below have smoothing parameters chosen by user. In the left panel one can see that when smoothing parameter is increased the curve tends to linearity and in the right panel there is the opposite change. What is interesting in automatic selection is that when data does not suggest any pattern only a regression line will be drawn. In `mgcv` -package it is actually ensured by adding to the smoothing parameter a small extra constant. The main thing in GAM estimation is the smoothing parameter selection but since it is also very extensive theoretically only the main parts of it will be represented.

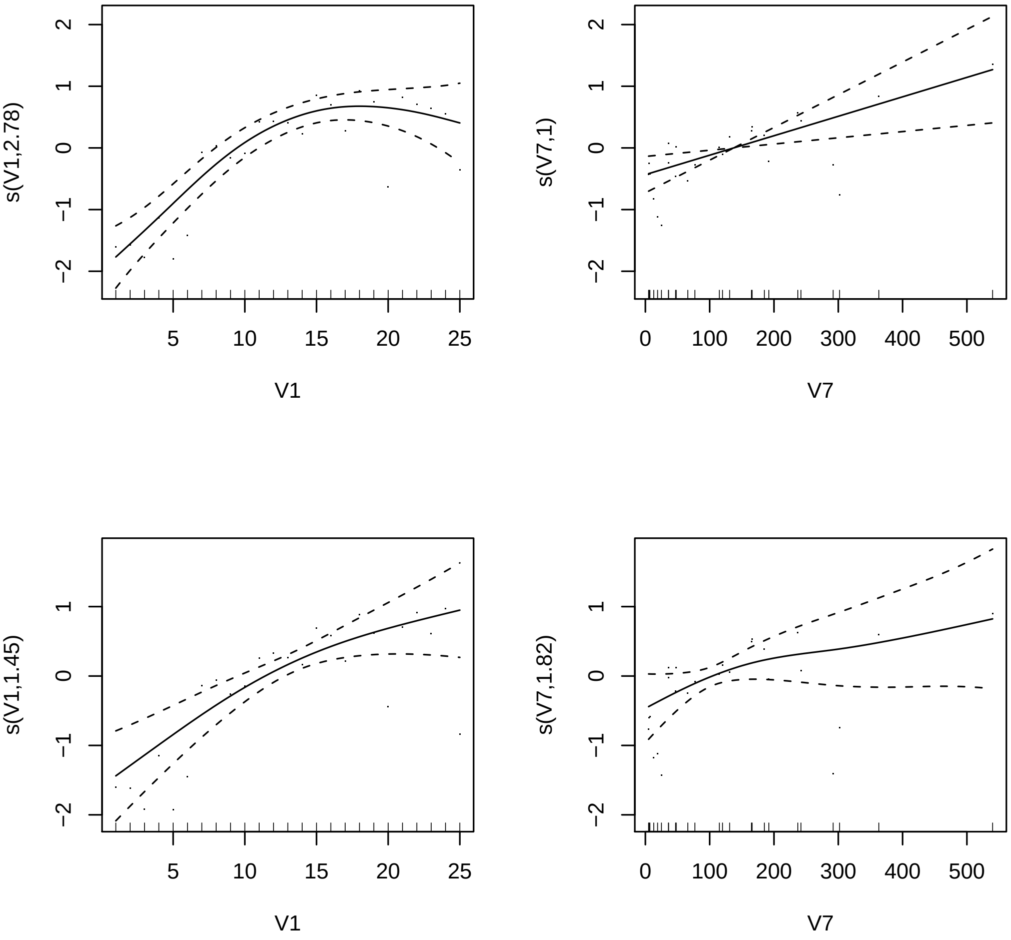

Figure 4.1. In the upper row smoothing parameters are chosen by GCV and below they are chosen to be 1.

4.2 Cross validation and UBRE

Cross validation is based on minimizing mean square prediction error. Since we do not usually have enough data to minimize prediction error with respect to new values (and if we had we would have to take a sample) we need to re-use the same data, i.e. predict in turn each one of observations with all the other observations. Since the data sets are usually small and with big data sets estimation could be slow the cross validation is a good way to choose a model that uses all the information available and fits well for new observations, if the data is a good representative of the population. Despite the fact that the ordinary cross validation is calculated from n models it may transformed to a faster form (see Wood 2006, Green and Silverman 1993 or Hastie and Tibshirani 1990) that gives us the possibility to calculate only one model

$$V_0 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}_i^{[-i]})^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \hat{\mu}_i)^2}{(1-H_{ii})^2}, \tag{4.2}$$

where $\hat{\mu}_i^{[-i]}$ is the predict of y_i calculated without y_i and H_{ii} is ith diagonal element of the hat matrix $\boldsymbol{H}$. In order to use cross validation method in GAM it has to be generalized as

$$V_g^p = \frac{n\left\|\boldsymbol{y} - \hat{\boldsymbol{\mu}}\right\|^2}{[n - \gamma\ tr(\boldsymbol{H})]^2}, \tag{4.3}$$

where γ is an additional constant that forces the fit to be a little bit smoother than it would be. In addition, the diagonal elements of the hat matrix are replaced by a trace of hat matrix. It has to be done since otherwise the diagonal elements would weight the observations unevenly. By orthogonal transformation we can rotate the hat matrix in a position where all the diagonal elements are even, which justifies multiplying nH_{ii}. That is simply the trace $tr(\boldsymbol{H})$. The rotation does not affect on parameters but it defines even proportions for observations in GCV. The weights and constraints of the fitting objective (4.1) need to be included in (4.3), after minimization parameters are transformed back to the original space. Minimizing is done by performance iteration or by outer iteration from which the first one is faster but suffers from occasional convergence problems. The biggest problem with GCV is the trace of the hat matrix that is slow to calculate.

A fact that we have an optional criterion $UBRE$ to use in smoothness selection does not really help much because it also involves calculation of the same trace as GCV. The difference between the two criterions is that $UBRE$ can be used only if the scale parameter of the model is assumed known, which is normally the case in binomial and poisson data. The aim of $UBRE$ is to minimize prediction error. Un-Biased Risk Estimator ($UBRE$) is defined as

$$V_u = \frac{1}{n}\left\|\boldsymbol{y} - \hat{\boldsymbol{\mu}}\right\|^2 + \frac{2}{n}\sigma^2\gamma\ tr(\boldsymbol{H}) - \sigma^2. \tag{4.4}$$

A little drawback in this theory is that GCV and $UBRE$ do not produce exactly same smoothing parameters. Luckily it is in any case often a subjective choise whether to use the automatic λ's or not. In some cases it might be better to adjust the smoothness a little. Both GCV and $UBRE$ are embeded in `mgcv` -package (multiple generalized cross validation). They are calculated within the Penalized Iteratively Re-weighted Least Squares procedure.

5 Inference

Inference in GAM context bases on the inference of GLM and standard linear modelling. Due to GAMs more general nature we can not form as exact results as in GLM and thus modelling with GAM has to be done with substantially more caution. A good basis for modelling is to assume that no smooth terms are needed and start with a GLM. Checking GLM results then might suggest trying some smooth terms instead. The question of how much data the smooth functions actually use is crucial on infering whether a parameter or a model is adequate.

5.1 Influence matrix and effective degrees of freedom

As with all the models that base on some linear transformation a GAM has also an explicit influence (hat) matrix that consists of linear explanatory variables and basis functions of smooth terms, including the smoothing parameters. Recall the hat matrix from standard linear regression

$$\boldsymbol{H_r} = \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'}. \tag{5.1}$$

It is easy to form an analogue from the number of parameters in regression to the number of parameters in GAM. As we know in regression the number of parameters is simply p but in GAMs model matrix there are both normal linear covariates and basis functions of smooth functions. In regression we see that the trace of $\boldsymbol{H_r}$ is $\boldsymbol{tr(H_r)} = \boldsymbol{tr(X(X'X)^{-1}X')} = \boldsymbol{tr((X'X)^{-1}X'X)} = \boldsymbol{tr(I)} = p$ and analogous to this we can define the degrees of freedom of a GAM as the trace of the hat matrix

$$\begin{aligned}
\boldsymbol{tr(H_g)} &= \boldsymbol{tr(X(X'WX+S)^{-1}X'W)} = \boldsymbol{tr((X'WX+S)^{-1}X'WX)} \\
&= \boldsymbol{tr((IX'WX+S(X'WX)^{-1}X'WX)^{-1}X'WX)} \\
&= \boldsymbol{tr(((I+S(X'WX)^{-1})X'WX)^{-1}X'WX)} \\
&= \boldsymbol{tr((I+S(X'WX)^{-1})^{-1}(X'WX)^{-1}X'WX)} \\
&= \boldsymbol{tr((I+S(X'WX)^{-1})^{-1})}.
\end{aligned}$$

Now it is straightforfard to notice that the trace is

$$\boldsymbol{tr(H_g)} = \sum_{i=1}^{n} \frac{1}{1+D_{ii}}, \tag{5.2}$$

where $\boldsymbol{D} = \boldsymbol{S}(\boldsymbol{X'WX})^{\boldsymbol{-1}}$. The single datums of this sum are so-called effective degrees of freedom (edf) of parameters. Thus for smooth functions we get the degrees of freedom by summing the corresponding diagonal elements and the whole trace is edf of the model. With edf it is possible to calculate p-values for single smooth terms and for changes in model deviance. Also the scale parameter ϕ can now be approximated as

$$\hat{\phi} = \frac{\sum_i V(\hat{\mu}_i)^{-1}(y_i - \hat{\mu}_i)^2}{n - tr(\boldsymbol{H_g})}, \tag{5.3}$$

that inherits from Pearson statistic X^2 (see (2.20)). Pearson statistic divided by the real value of scale parameter is considered to be a sum of standard normal sample squared, i.e. chi-square distributed with $n - p = n - tr(\boldsymbol{H_g})$ degrees of freedom. Thus its expectation is also $n - tr(\boldsymbol{H_g})$ and solving $X^2/\phi = n - tr(\boldsymbol{H_g})$ for ϕ yields the result.

Effective degrees of freedom increase as we build more and more wiggly smooth functions. In Figure 4.1 one can see how edf changes in the left panel from 2.78 to 1.45 and in the right panel from 1 to 1.82. The upper values are due to automatic smoothing parameter selection and thus they are in mathematical sense quite optimal.

5.2 Covariance matrix and approximate p-values

Now that we have means to estimate how much data we have used for single parameter estimate we can find a way to calculate approximate, but somewhat satisfyingly exact, p-values. Similarly as in standard linear model and GLM we start by defining a covariance matrix of parameters. As the parameter vector in GAM is $\hat{\boldsymbol{\beta}} = (\boldsymbol{X'WX} + \boldsymbol{S})^{\boldsymbol{-1}}\boldsymbol{X'Wy}$ and since $\mathrm{Cov}(\boldsymbol{Ax}) = \boldsymbol{A}\,\mathrm{Cov}(x)\boldsymbol{A'}$ we have

$$\begin{aligned}
\mathrm{Cov}(\hat{\boldsymbol{\beta}}) = \boldsymbol{V_e} &= (\boldsymbol{X'WX} + \boldsymbol{S})^{\boldsymbol{-1}}\boldsymbol{X'W}\,\mathrm{Cov}(y)\boldsymbol{WX}(\boldsymbol{X'WX} + \boldsymbol{S})^{\boldsymbol{-1}} \\
&= (\boldsymbol{X'WX} + \boldsymbol{S})^{\boldsymbol{-1}}\boldsymbol{X'WW}^{\boldsymbol{-1}}\boldsymbol{WX}(\boldsymbol{X'WX} + \boldsymbol{S})^{\boldsymbol{-1}}\phi \\
&= (\boldsymbol{X'WX} + \boldsymbol{S})^{\boldsymbol{-1}}\boldsymbol{X'WX}(\boldsymbol{X'WX} + \boldsymbol{S})^{\boldsymbol{-1}}\phi.
\end{aligned}$$

From these results we get an approximation that $\hat{\boldsymbol{\beta}} \sim N(E(\hat{\boldsymbol{\beta}}), \boldsymbol{V_e})$, where $E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$ only when $\boldsymbol{\beta} = \boldsymbol{0}$. So this result is applicape only for calculating p-values of parameters. Let $\boldsymbol{V_{\beta_j}}$ denote the part of the covariance matrix that corresponds to parameter vector $\boldsymbol{\beta_j}$ of a smooth function. Under null hypotesis $\boldsymbol{\beta_j} = \boldsymbol{0}$ we have (see Wood 2006 for the hat matrix)

$$\hat{\boldsymbol{\beta}}'_{\boldsymbol{j}} \boldsymbol{V}^{r-}_{\hat{\boldsymbol{\beta}}_{\boldsymbol{j}}} \hat{\boldsymbol{\beta}}_{\boldsymbol{j}} \sim \chi^2_r, \tag{5.4}$$

where $r = rank(V_{\hat{\boldsymbol{\beta}}_{\boldsymbol{j}}^{r-}})$. The rank of covariance matrix is used instead of dimension because heavy penalization may shrink also the covariance matrix. If the scale parameter is unknown F-test will place again with an estimate of scale parameter plugged in. If the smoothing parameters are estimated, which they usually are, these p-values are not exact since smoothing parameter selection brings an extra amount of uncertainty to modelling. In practice p-values might be as small as half of the real values but still a very clear result is reliable.

5.3 Bayesian confidence intervals

Bayesian inference on GAM is based on the fact that using penalization is using prior beliefs of what we think what might lead us to the right amount of smoothness. Defining a prior distribution for $\boldsymbol{\beta}$ would already lead us to sufficient confidence intervals when inspecting the whole model but this would not work well enough for separate model terms (Wood 2006). Instead Wood suggests using a joint posterior density of both $\boldsymbol{\beta}$ and $\hat{\boldsymbol{\lambda}}$

$$f(\boldsymbol{\beta}, \hat{\boldsymbol{\lambda}}|\boldsymbol{y}) = f(\boldsymbol{\beta}|\hat{\boldsymbol{\lambda}}, \boldsymbol{y})f(\hat{\boldsymbol{\lambda}}|(y)) \approx f(\boldsymbol{\beta}|\hat{\boldsymbol{\lambda}}, \boldsymbol{y})f_{\hat{\lambda}}(\hat{\boldsymbol{\lambda}}), \tag{5.5}$$

where $f_{\hat{\lambda}}(\hat{\boldsymbol{\lambda}})$ is a bootstrap sampling distribution of $\hat{\boldsymbol{\lambda}}$ ($f(\hat{\boldsymbol{\lambda}}|(y))$ is unknown). This modification takes into a count also the selection of smoothing parameters. Simulation is simple as we take some N bootstrap samples ($\boldsymbol{y}^{[k]}$'s) from $\boldsymbol{y}$ and fit the same model for each sample. From those models then $\hat{\boldsymbol{\lambda}}^{[k]}$'s are stored and with each set of new smoothing parameters a model for the original data $\boldsymbol{y}$ is calculated and $\hat{\boldsymbol{\beta}}^{[k]}$'s and $\boldsymbol{V}_{\boldsymbol{\beta}}^{[k]}$'s are stored. After having calculated N different distributions $\boldsymbol{\beta} \sim N(\boldsymbol{\beta}, \boldsymbol{V}_{\beta})$ it is straightforward to randomly choose from those distributions and take one random sample from the ones chosen. Based on the sampling distribution of $\boldsymbol{\beta}$ we get the confidence intervals.

5.4 Model comparison and selection

As Hastie and Tibshirani (1990) presented theory on generalized additive models they suggested a systematic approach on model selection since automatic selection of smoothing parameters had not been solved yet. They simply proposed testing each variable with three different preset degrees of freedom. They also modified their *Backfitting algorithm* to be adaptive in a way that iteration would lead to smaller *GCV* values. Even though automatic selection is now available model selection, at least selecting which terms to include, has to be done with caution.

The basic methods of model comparison are mostly the same as in GLM. Residuals, fitted values and observed values need to be compared pairwise in order to ensure that the model is adequate. Also normal quantile-quantile plots are available for GAM. In addition to GLM smooth terms need to be evaluated carefully. Wood (2006) suggests checking whether a little bit more penalized model could do as well as the one chosen by *GCV*. This can be done by adjusting the gamma-parameter in *GCV* or *UBRE* (see (4.3) and (4.4)). Sometimes checking if the basis dimension or the basis itself of the smoother is right might help if a smooth function does not appear to adjust nicely. That is changing the smoother or its number of knots.

A numerical test for comparing two GAMs is done by chi-square test of F-test just as in GLM. That is

$$\lambda = 2\left[l(\hat{\eta}_1) - l(\hat{\eta}_0)\right] \sim \chi^2_{EDF_1 - EDF_0}, \tag{5.6}$$

where EDFs are effective degrees of freedom. Comparing two models this way has some restrictions though. One has to ensure that each term in the null model has same or smaller effective degrees of freedom than the same term in alternative model.

6 Analysing data with GAM

The data in our example is a part of a longitudinal study that was conducted to explain mothers depression symptoms and their effect on adolescents well-being and socio-emotional development. The initial target group of the study was all women who were waiting their first child in Tampere in 1989-1990. Around 90% of the target group i.e. 349 mothers participated at the first stage of the study. Drop-outs during the study are considered to be random, although boys' mothers left more than girls' mothers. At the last stage of study there were still 198 mothers and 90 mothers participated each time.

6.1 Longitudinal EPDS–variable

The study was conducted in four different stages and in seven data collection points which measure for example mothers depression symptoms and adolescents socio-emotional problems in certain phases of adolescents development. The first stage includes the first four data collection points $T1 - T4$ with $T1$ measured during the last trimester of pregnancy, $T2$ left out from this study and $T3$ and $T4$ measured two and six months after childbirth. The second phase i.e. measurement $T5$ was made when the adolescent was 5–6 years old and about to start preschool. Measurements at $T6$ and $T7$ (stages three and four) represent the age of 7–8 years when adolescent had started school and age of 16, which is the last study point.

Table 6.1. Frequencies of classified EPDS scores (0:normal 1:subclinical or clinical)

Variable	T1	T3	T4	T5	T6	T7
EPDS						
0	147	128	125	123	114	140
1	43	26	30	34	27	36
missing	8	44	43	41	57	22

The measurements at all study stages were produced by simple questionnaires. Mothers depression symptoms were measured by Edinburgh Postnatal Depression Scale (EPDS). EPDS -questionnaire consists of ten questions all of them having four options scored 0–3. So calculating a sum score gives a scale of 0–30. In EPDS score a limit for clinical depression is considered to be 13 points and 10 for subclinical, below 10 being normal. The frequencies of EPDS are in table 6.1. Other statistics considering the mothers and their families are described in table 6.2.

Table 6.2. Statistics of EPDS -data

Variable	abs.	%
Gender		
0 boy	92	46,5
1 girl	106	53,5
SES		
0 lower	81	40,9
1 higher	117	59,1
Mothers education		
0 elementary school	71	37,0
1 high school or higher	121	63,0
missing	6	
Mothers marital status		
0 cohabitation or marriage	158	82,3
1 single parent	34	17,7
missing	6	
Number of children		
0 1	25	13,1
1 >1	166	88,9
missing	7	

The adolescents emotional and behavioral problems were measured at the same time as the last three measurements of the mothers. At stages $T5 - T7$ mothers filled out Child Behavioral Checklist CBCL and in addition at $T7$ adolescents filled out Youth Self Report YSR (see Appendix A). Scores produced by CBCL and YSR are divided in internalizing and externalizing problem scores and also social competence is measured by the answers. Internalizing symptoms mostly describe depression and anxiety type symptoms and externalizing symptoms consist more of behavioral symptoms. We will be concentrating on the CBCL and YSR in modelling.

6.2 Summarizing EPDS–variable by mixed model

Since our goal is to model CBCL and YSR internalizing and externalizing problem scores we need to find a way to summarize the longitudinal EPDS -variable. Doing this ensures that we can use the mothers depression symptoms to predict the adolescents problems. One could just use the separate EPDS -scores from different measurements and see if some of them stand out but instead we will try to simplify the problem a little bit. A linear mixed model is a natural choise to use because it is simple and understandable. A mixed model produces random effects for each mother with respect to time and thus summarizes all the information in an interpretable way. Below is the R-result from a mixed model that has all the statistically significant predictors. In the model *vuosi* represents time, $T1$ mother's EPDS score during the pregnancy, *nrchild* the number of children in the family, *edu* mother's education and *age* is mother's age.

```
> sm <- lme(EPDS~T1*vuosi+factor(nrchild)+factor(edu)+age,
+ data=datalme2, random=~1+vuosi+I(vuosi^2), na.action=na.omit,
+ method="ML")
> summary(sm)
Linear mixed-effects model fit by maximum likelihood
 Data: datalme2
       AIC      BIC    logLik
  4083.171 4147.531 -2027.585

Random effects:
 Formula: ~1 + vuosi + I(vuosi^2) | nro
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev     Corr
(Intercept) 2.99890759 (Intr) vuosi
vuosi       1.22647359 -0.767
I(vuosi^2)  0.07789046  0.748 -0.993
Residual    2.85160206

Fixed effects: EPDS~T1*vuosi+factor(nrchild)+factor(edu) + age
                      Value Std.Error  DF   t-value p-value
(Intercept)       -3.391035 2.3319849 548 -1.454141  0.1465
T1                 0.421472 0.0528728 178  7.971427  0.0000
vuosi              0.162723 0.0414406 548  3.926668  0.0001
factor(nrchild)2   1.072880 0.5634457 178  1.904141  0.0585
factor(edu)2      -1.665393 0.3917595 178 -4.251059  0.0000
age                0.146489 0.0480342 178  3.049677  0.0026
T1:vuosi          -0.021479 0.0051852 548 -4.142385  0.0000
...
Standardized Within-Group Residuals:
       Min         Q1        Med          Q3        Max
-2.7720640 -0.5257349 -0.1240126  0.4784764  3.5362153

Number of Observations: 733
Number of Groups: 183
```

In the mixed model *sm* we have random effects for intercept, time and time squared. By taking all the significant predictors into a count we eliminate their effect from the random effects. These random effects can be used when modelling internalizing or externalizing problem scores of CBCL and YSR. This setting allows us to see if the level or tendency of the mothers' depression symptoms has some effect on the adolescents problems.

6.3 Modelling Externalizing symptoms with GAM

We will now try to model YSR (at time point $T7$) externalizing problem scores, i.e. adolescents' own opinion of their externalizing symptoms, with GAM. Taking a look at the variable $next7$ plotted against each possible predictor in Figure 6.1 we immediately see that some of the predictors are good candidates while some others do not show any pattern at all. Variables $nint7$, $aext7$, $aint7$ and $itot7$ seem to have some pattern in the plots but in the first model we set all the variables in the model and the result is shown below.

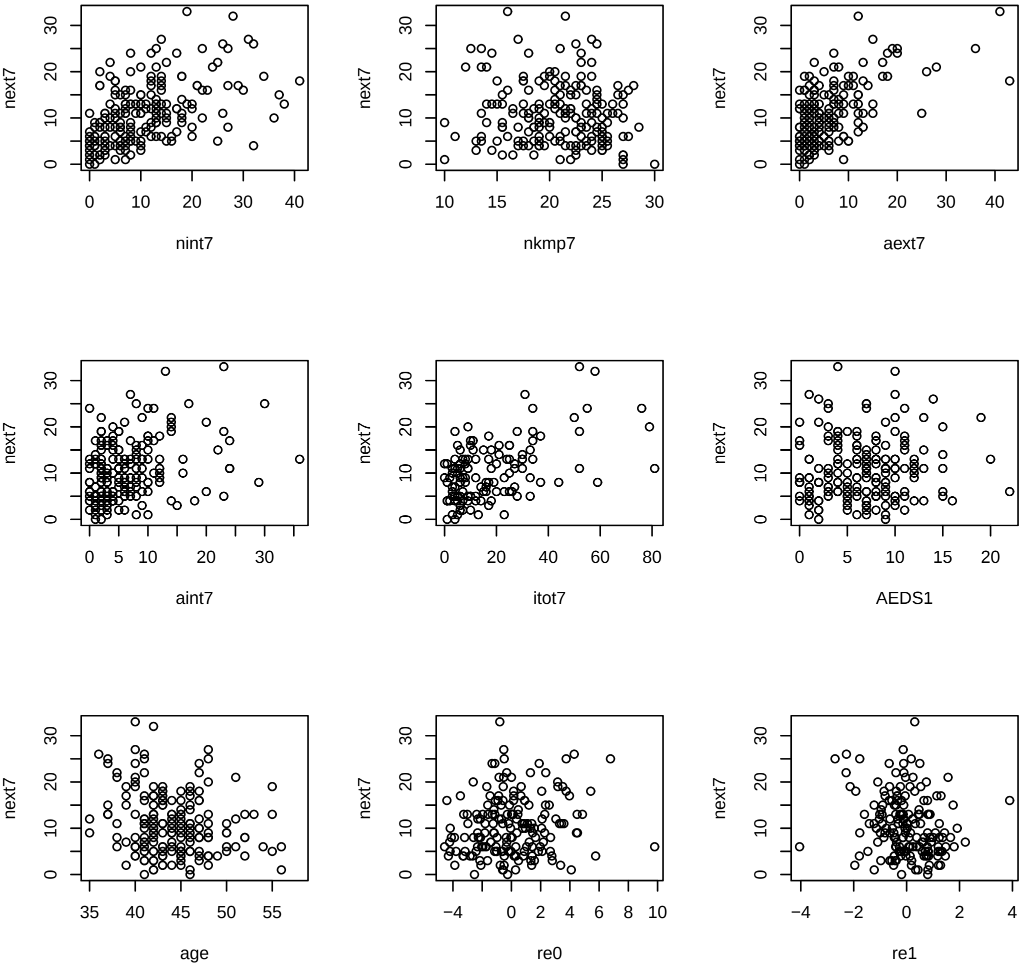

Figure 6.1. Variable $next7$ plotted against all the predictors

The choises of logistic link function ang gamma distribution are easily justified by inspecting the histogram of $next7$ in the Figure 6.2. It is not perfect but by trying every possible combination one can accept such choise. And it works nicely. The results in model $gam1$ are very much expected and those variables that were strong candidates beforehand are statistically significant.

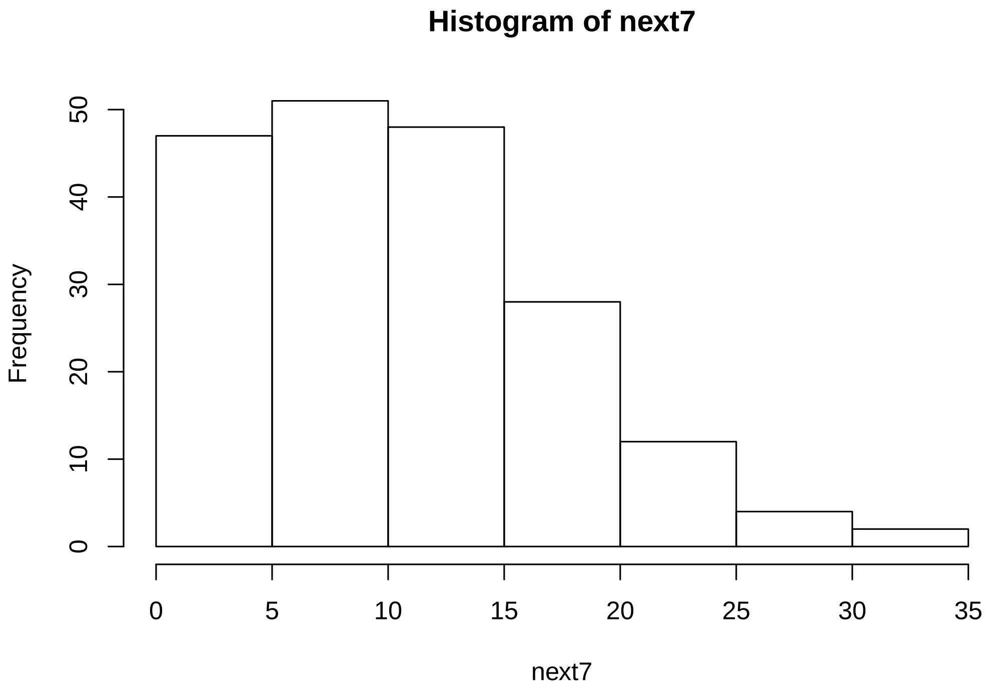

Figure 6.2. Histogram of variable $next7$.

There is a plenty of ways which to choose in this situation and possibly not one is a better option than the others. But since the plots clearly predict the significance it is justifiable to include the variables that first stand out. Furthermore the R-result shows adjusted R-squared, explained deviance, estimated *GCV* score and sample size used. Basically this fit is simply a GLM since there is no smooth functions used. It is easier to fisrt search for parametric terms and then see if some of them or some other term can be a better predictor as a smooth function, inspecting plots again. Now the factor covariates do not show any significance but they should be tested again when the model changes.

```
> summary(gam1)

Family: Gamma
Link function: log

Formula:
next7 ~ nint7 + nkmp7 + aext7 + aint7 + itot7 + AEDS1 + age +
    re0 + re1 + factor(nrchild) + factor(edu) + factor(mstatus) +
    factor(sex) + factor(ses)

Parametric coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       2.748891   0.689395   3.987 0.000127 ***
nint7             0.033992   0.006998   4.857 4.41e-06 ***
```

```
nkmp7            -0.010602   0.013555  -0.782 0.435996
aext7             0.044735   0.011023   4.058 9.82e-05 ***
aint7            -0.040265   0.011189  -3.599 0.000499 ***
itot7             0.006481   0.003883   1.669 0.098283 .
AEDS1             0.006819   0.012143   0.562 0.575705
age              -0.014619   0.011595  -1.261 0.210309
re0              -0.026687   0.030922  -0.863 0.390181
re1              -0.001766   0.080915  -0.022 0.982632
factor(nrchild)2 -0.164495   0.174245  -0.944 0.347421
factor(edu)2      0.085307   0.113526   0.751 0.454161
factor(mstatus)2  0.243210   0.263830   0.922 0.358829
factor(sex)2      0.084297   0.103425   0.815 0.416979
factor(ses)2     -0.079345   0.107259  -0.740 0.461182
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1


R-sq.(adj) =  0.428   Deviance explained = 45.3%
GCV score = 0.25622  Scale est. = 0.2228    n = 115
```

Now taking out the insignificant terms and continuing with only variables *nint*7, *aext*7, *aint*7 and intercept we get a simpler model *gam*1*a* which is pretty much the same as the first one. The estimates of the parameters of course change a little and the deviance explained is smaller too. This model seems to be a reasonable start.

```
> summary(gam1a)
...
Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.929389   0.058988  32.708  < 2e-16 ***
nint7        0.032777   0.004829   6.787 1.58e-10 ***
aext7        0.044567   0.005946   7.496 2.83e-12 ***
aint7       -0.023065   0.007602  -3.034  0.00277 **
...
R-sq.(adj) =  0.349   Deviance explained = 38.4%
GCV score = 0.22448  Scale est. = 0.21962   n = 185
```

From Figure **??** we see that variable *nint*7 might have some curvature and testing this intuition by analysis of deviance ensures it. Though we are using the gam-function we are still modelling GLMs and thus analysis of deviance gives quite reliable results.

```
Analysis of Deviance Table

Model 1: next7 ~ nint7 + aext7 + aint7
Model 2: next7 ~ nint7 + I(nint7^2) + aext7 + aint7
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1       181     39.752
```

```
2       180     37.558  1    2.194  0.001184 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Now what a GAM can do in this situation is testing whether we could replace $nint7 + I(nint7^2)$ by a smooth function of $nint7$. Since GAM can find any function structure we will see if we really should have a more complex function or if the parametric structure shows all the pattern there is. Now in a GAM $gam1c$ we have one smooth function $s(nint7)$ that has 3.213 degrees of freedom while the parametric structure before had 1 + 1, of course. The p-value is practically zero so the model seems nice. Although testing the difference of these two models by the analysis of deviance gives a p-value 0.03461, which is not very ensuring result. Then again the deviance explained by the latter model $gam1c$ is 43.4% while in $gam1b$ it was 41.8%.

```
> summary(gam1c)
...
Formula:
next7 ~ s(nint7) + aext7 + aint7

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.251277   0.054434  41.358  < 2e-16 ***
aext7        0.044579   0.005791   7.698 9.04e-13 ***
aint7       -0.020520   0.007431  -2.761  0.00636 **
...
Approximate significance of smooth terms:
           edf Ref.df     F  p-value
s(nint7) 3.213  3.997 16.17 2.57e-11 ***
...
R-sq.(adj) =  0.368   Deviance explained = 43.4%
GCV score = 0.21146  Scale est. = 0.20435   n = 185
```

Continuing modelling with a similar testing of the variable $aext7$ by replacing it with $aext7+I(aext7^2)$ shows that there is some curvature, with p-values below 0.01. Furthermore trying to replace $aext7+I(aext7^2)$ with a smooth function of $aext7$ similarly as before results as an almost zero change in deviance. Actually residual deviance even increases and thus the parametric form is better. Again similarly testing variable $aint7$ shows that neither a form $aint7+I(aint7^2)$ nor a smooth function of $aint7$ could be used.

```
> summary(gam1d)
...
Formula:
next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7

Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.1712362  0.0613870  35.370  < 2e-16 ***
```

```
aext7        0.0733160  0.0122305   5.995 1.11e-08 ***
I(aext7^2)  -0.0009745  0.0003437  -2.835  0.00511 **
aint7       -0.0229126  0.0073621  -3.112  0.00216 **
...
Approximate significance of smooth terms:
           edf Ref.df     F  p-value
s(nint7) 3.402   4.22 16.04 1.21e-11 ***
...
R-sq.(adj) =   0.52   Deviance explained = 45.8%
GCV score = 0.20512  Scale est. = 0.19691   n = 185

> anova(gam1c,gam1d,test="Chi")
Analysis of Deviance Table

Model 1: next7 ~ s(nint7) + aext7 + aint7
Model 2: next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7
  Resid. Df Resid. Dev     Df Deviance P(>|Chi|)
1    178.79     36.536
2    177.60     34.971 1.1881   1.5649  0.006586 **

> summary(gam1e)
...
Formula:
next7 ~ s(nint7) + s(aext7) + aint7

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.525929   0.058495  43.182  < 2e-16 ***
aint7       -0.022101   0.007357  -3.004  0.00305 **
...
Approximate significance of smooth terms:
           edf Ref.df     F  p-value
s(nint7) 3.351  4.157 16.26 1.16e-11 ***
s(aext7) 2.096  2.592 24.14 6.57e-12 ***
...
R-sq.(adj) =  0.519   Deviance explained = 45.7%
GCV score = 0.20579  Scale est. = 0.1975    n = 185

Analysis of Deviance Table

Model 1: next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7
Model 2: next7 ~ s(nint7) + s(aext7) + aint7
  Resid. Df Resid. Dev       Df  Deviance P(>|Chi|)
1    177.60     34.971
2    177.55     35.067 0.046102 -0.096131
```

After testing all the remaining variables, continuous and categorical, there is only one more significant parameter to add. The number of children, *nrchild*, in the adolescents family seems to have a protective effect on externalizing symptoms. Our model *gam1h* now explains 46.6% of the deviance and R-squared indicates a little bit better result with a value of 0.52.

```
> summary(gam1h)
...
Formula:
next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7
+ factor(nrchild)

Parametric coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       2.3486834  0.1051798  22.330  < 2e-16 ***
aext7             0.0737127  0.0121521   6.066 7.86e-09 ***
I(aext7^2)       -0.0009371  0.0003404  -2.753 0.006523 **
aint7            -0.0245764  0.0073351  -3.351 0.000988 ***
factor(nrchild)2 -0.2075395  0.0978887  -2.120 0.035399 *
...
Approximate significance of smooth terms:
           edf Ref.df     F  p-value
s(nint7) 3.601  4.458 16.29 3.22e-12 ***
...
R-sq.(adj) =   0.52   Deviance explained = 46.6%
GCV score = 0.20111  Scale est. = 0.19171   n = 184
```

With an R-command `gam.check` we get model checking plots for a GAM. Now in Figure 6.3 we have Normal Q-Q plot, residuals against predicted values, histogram of residuals and predicted values against observed values. Taking a look at the Q-Q plot first we can observe that the quantiles are not perfect but somewhat satisfying. Then observing residuals against predicted values we take notion that residual variance seems to decrease as predicted values increase, which is not as bad as the opposite situation would be. It might tell us that, since there is only a few really big data values, the parameter estimates could be a little over estimated. Then again predicted values are quite satisfying compared to observed values, small and large. One could say that predictions in smaller values are not satisfying at all but in fact it could be very hard to get more accurate, and reliable, predictions as we are modelling a psychological phenomenom measured by questionnaires. The histogram of residuals shows some skewness though but in a tolarable scale.

What comes to the selected model *gam1h* it is easy to see how the parametric terms effect linear predictor but the nonparametric term has to be calculated and drawn by program. We can draw it in three dimensions with another explanatory variable by R-command `vis.gam`. In Figure 6.4 we have variables *nint7* and *aext7* plotted against the linear predictor and as we can see the variables seem to have some understandable correlation between the predictors. As adolescent reports more internalizing problems also the externalizing problems are bigger and it seems that mother can detect adolescent's externalizing prob-

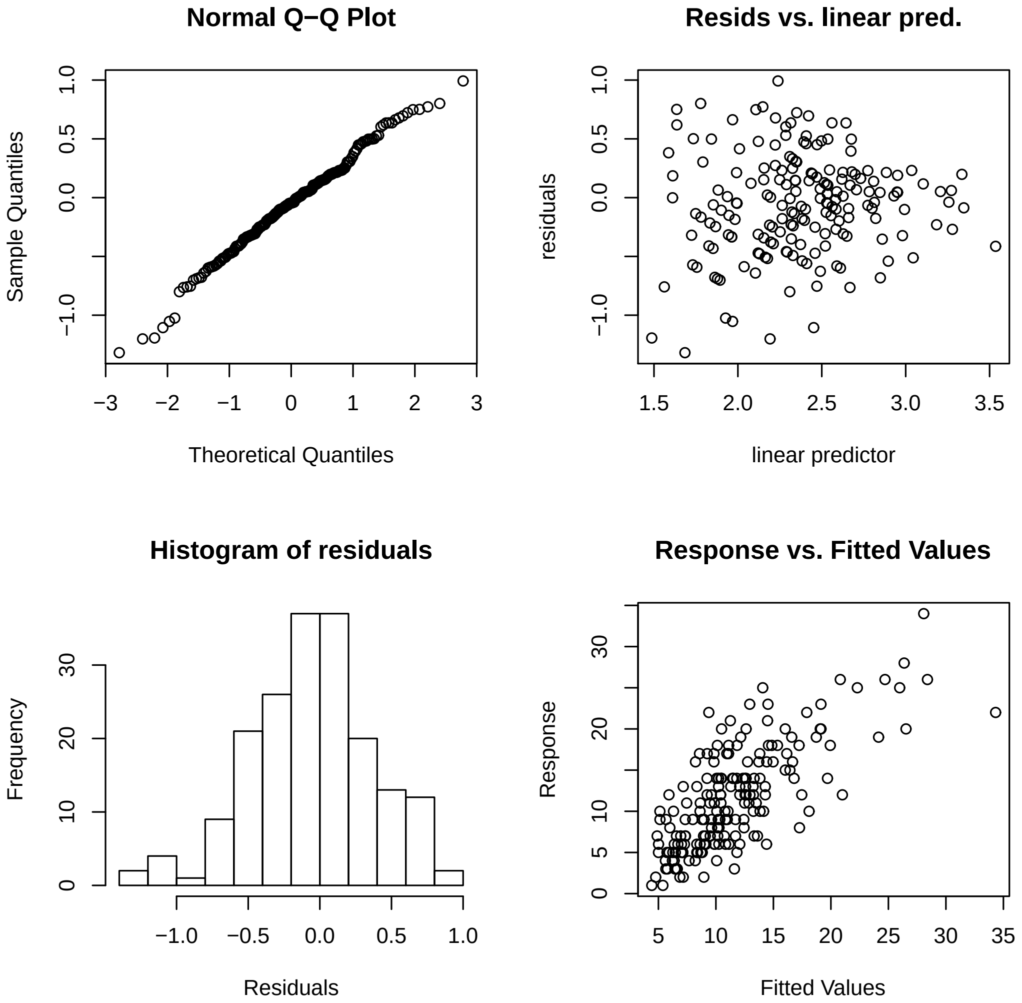

Figure 6.3. Model check for model *gam1h*.

lems. But the variable *aint7* does actually have a slightly negative effect on the linear predictor, which is a little surprising.

When interpreting the parameter estimates one has to remember that our model is based on a logistic link function which changes things. When a variable increases by 1 and if the parameter estimate is 0.25 the expected value is multiplied by $e^{0.25} = 1.284$. So the model is of the form $\mu = e^{\sum \beta_j x_j}$. So as we have 2.349 as intercept the basic level of externalizing symptoms is $e^{2.349} \approx 10.5$.

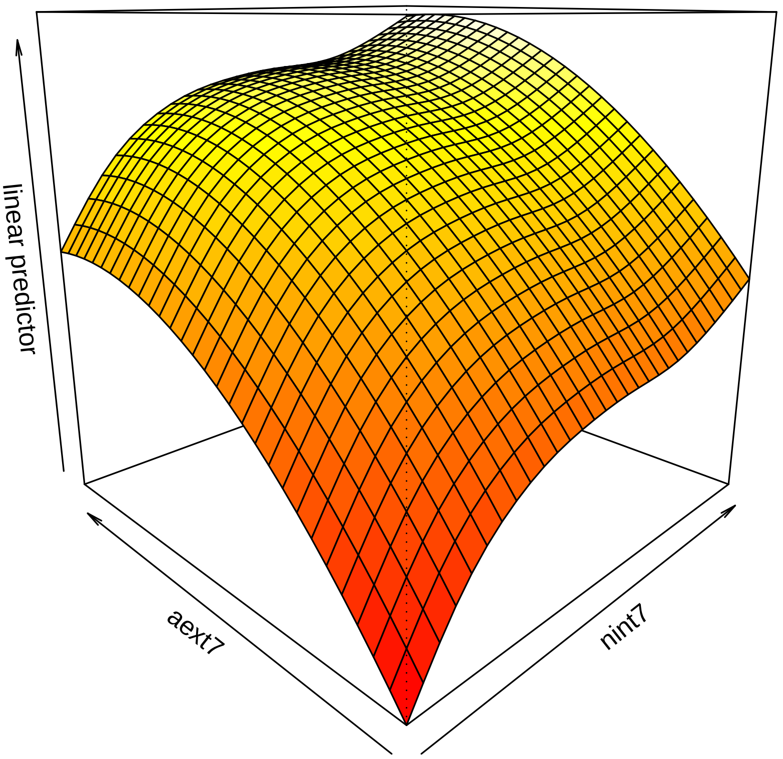

Figure 6.4. A three dimensional view on variables *nint*7 and *aext*7 plotted against the linear predictor.

7 Conclusions

As seen in the previous chapter generalized additive model can be used as an assisting tool for model building and when necessary a GAM might be the right choise to make when choosing the final model, as it turned out in our example. Allthough, we could have chosen to stay within the generalized linear model framework. It would not have been a totally wrong desicion but in a way a GAM represents the uncertainty of the described situation. And in fact the chosen model is not that different from parametric form, as the smooth function is quite simple.

Of course, the presented example shows only one way to use GAMs. There are obviously many situations which can not be modelled with a purely parametric model. As an example one could mention data that have some kind of change point in their behavior when a certain predictor exceeds a limiting value. Such situations provide more concrete usage for GAMs as the smooth functions do not only point out the non-parametric form of the data but work with less data and better accuracy.

After having checked out the underlying theory, a generalized additive model is not much more complicated to use than a generalized linear model. One just has to be cautious not to model noise or over interpret modelled smooth functions. The R-function by Wood (2006) seems to never fail and is rather fast for medium sized data sets. After all generalized additive models are definitely worth of trying if simpler techniques fail to explain enough. Or in case one just wants to be sure for not missing any patterns in the data.

Working with generalized linear models, nonparametric regression methods and generalized additive models have been very taughtful and given me some perspective after years of watching the world through parametric frames. The mathematical theory behind nonparametric results were perhaps the biggest challenge in the quest for understanding GAM thoroughly. Searching through the extensive theory with no help would not have been easy without good guidance.

Bibliography

de Boor, C. (2001), A practical guide to splines (2nd ed.), *Applied Mathematical Sciences*, New York : Springer.

Eubank, R. L. (1988), Spline smoothing and nonparametric regression, New York : Dekker.

Green, P.J. & Silverman, B.W. (1994), Nonparametric Regression and Generalized Linear Models, *Monographs on Statistics and Applied Probability*, London: Chapman & Hall.

Hastie, T. J. & Tibshirani, R. J. (1986), Generalized Additive Models, *Statistical Science*, 1, pp. 297-310.

Hastie, T. J. & Tibshirani, R. J. (1990), Generalized Additive Models, *Monographs on Statistics and Applied Probability*, London: Chapman & Hall.

McCullagh, P. & Nelder, J. A. (1989), Generalized Linear Models (2nd ed.), *Monographs on Statistics and Applied Probability*, London: Chapman & Hall.

Wahba, G. (1990), Spline models for observational data, Philadelphia (Penn.): Society for Industrial and Applied Mathematics.

Wood, S. N. (2003), Thin Plate Regression Splines, *Journal of the Royal Statistical Society. Ser. B*, 65, pp. 95–114.

Wood, S. N. (2006), Generalized Additive Models: An Introduction with R, *Texts in Statistical Science*, Boca Raton: Chapman & Hall/CRC.

Wu, H. & Zhang, J. T. (2006), Nonparametric Regression Methods for Longitudinal Data Analysis: Mixed-Effects Modeling Approaches, *Wiley Series in Probability and Statistics*, Hoboken (N.J.) : Wiley-Interscience.

Appendix A

List of variables in the EPDS -data

Variable	min	median	mean	max	NAs
next7	0.00	10.00	10.99	33.00	6
nint7	0.00	8.50	10.81	41.00	6
nkmp7	10.00	21.25	20.79	30.00	10
aext7	0.00	4.00	6.02	43.00	7
aint7	0.00	5.00	6.57	36.00	7
itot7	0.00	12.00	18.49	81.00	71
AEDS1	0.00	7.00	6.721	22.00	8
age	35.00	43.00	44.04	58.00	0
re0	-4.57	-0.28	0.01	9.80	15
re1	-4.05	-0.06	-0.00	3.89	15

Variable	explanation
next7	YSR externalizing problem score
nint7	YSR internalizing problem score
nkmp7	YSR social competence score
aext7	mother's CBCL externalizing symptom score
aint7	mother's CBCL internalizing symptom score
itot7	father's CBCL total score
AEDS1	mother's EPDS score during the pregnancy
age	mother's age at time point $T7$
re0	mixed model random effect for intercept
re1	midex model random effect for slope
edu	mother's education
nrchild	number of children in the family
mstatus	mother's marital status
sex	adolescent's gender
ses	family's socioeconomic status

Appendix B

R code and results used in analyses

```
> gam1 <- gam(next7 ~ nint7+nkmp7+aext7+aint7+itot7+
+ AEDS1+age+re0+re1+factor(nrchild)+factor(edu)+
+ factor(mstatus)+factor(sex)+factor(ses),
+ family=Gamma(link="log"), data=redata2b)
> summary(gam1)

Family: Gamma
Link function: log

Formula:
next7 ~ nint7 + nkmp7 + aext7 + aint7 + itot7 + AEDS1 + age +
    re0 + re1 + factor(nrchild) + factor(edu) + factor(mstatus) +
    factor(sex) + factor(ses)

Parametric coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       2.748891   0.689395   3.987 0.000127 ***
nint7             0.033992   0.006998   4.857 4.41e-06 ***
nkmp7            -0.010602   0.013555  -0.782 0.435996
aext7             0.044735   0.011023   4.058 9.82e-05 ***
aint7            -0.040265   0.011189  -3.599 0.000499 ***
itot7             0.006481   0.003883   1.669 0.098283 .
AEDS1             0.006819   0.012143   0.562 0.575705
age              -0.014619   0.011595  -1.261 0.210309
re0              -0.026687   0.030922  -0.863 0.390181
re1              -0.001766   0.080915  -0.022 0.982632
factor(nrchild)2 -0.164495   0.174245  -0.944 0.347421
factor(edu)2      0.085307   0.113526   0.751 0.454161
factor(mstatus)2  0.243210   0.263830   0.922 0.358829
factor(sex)2      0.084297   0.103425   0.815 0.416979
factor(ses)2     -0.079345   0.107259  -0.740 0.461182
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1


R-sq.(adj) =  0.428   Deviance explained = 45.3%
```

```
GCV score = 0.25622  Scale est. = 0.2228    n = 115
> gam.check(gam1)

Method: GCV   Optimizer: outer newton
Model required no smoothing parameter selection
> plot(gam1,residuals=T,all.terms=T)
> gam1a <- gam(next7 ~ nint7+aext7+aint7, family=Gamma(link="log"),
+ data=redata2b)
> summary(gam1a)

Family: Gamma
Link function: log

Formula:
next7 ~ nint7 + aext7 + aint7

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.929389   0.058988  32.708  < 2e-16 ***
nint7        0.032777   0.004829   6.787 1.58e-10 ***
aext7        0.044567   0.005946   7.496 2.83e-12 ***
aint7       -0.023065   0.007602  -3.034  0.00277 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1


R-sq.(adj) =  0.349   Deviance explained = 38.4%
GCV score = 0.22448  Scale est. = 0.21962   n = 185
> gam.check(gam1a)

Method: GCV   Optimizer: outer newton
Model required no smoothing parameter selection
> plot(gam1a,residuals=T,all.terms=T)
>
> gam1b <- gam(next7 ~ nint7+I(nint7^2)+aext7+aint7,
+ family=Gamma(link="log"), data=redata2b)
> summary(gam1b)

Family: Gamma
Link function: log

Formula:
next7 ~ nint7 + I(nint7^2) + aext7 + aint7

Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.7612713  0.0746208  23.603  < 2e-16 ***
```

```
nint7        0.0662754  0.0111860   5.925 1.56e-08 ***
I(nint7^2)  -0.0011048  0.0003374  -3.274  0.00127 **
aext7        0.0429615  0.0058267   7.373 5.86e-12 ***
aint7       -0.0194458  0.0074828  -2.599  0.01013 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1


R-sq.(adj) =  0.375   Deviance explained = 41.8%
GCV score = 0.21445  Scale est. = 0.20865   n = 185
> gam.check(gam1b)

Method: GCV   Optimizer: outer newton
Model required no smoothing parameter selection
> anova(gam1a,gam1b,test="Chi")
Analysis of Deviance Table

Model 1: next7 ~ nint7 + aext7 + aint7
Model 2: next7 ~ nint7 + I(nint7^2) + aext7 + aint7
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1       181     39.752
2       180     37.558  1    2.194  0.001184 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> plot(gam1b,residuals=T,all.terms=T)
>
> gam1c <- gam(next7 ~ s(nint7)+aext7+aint7,
+ family=Gamma(link="log"), data=redata2b)
> summary(gam1c)

Family: Gamma
Link function: log

Formula:
next7 ~ s(nint7) + aext7 + aint7

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.251277   0.054434  41.358  < 2e-16 ***
aext7        0.044579   0.005791   7.698 9.04e-13 ***
aint7       -0.020520   0.007431  -2.761  0.00636 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
           edf Ref.df     F  p-value
s(nint7) 3.213  3.997 16.17 2.57e-11 ***
```

```
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.368   Deviance explained = 43.4%
GCV score = 0.21146  Scale est. = 0.20435   n = 185
> gam.check(gam1c)

Method: GCV   Optimizer: outer newton
full convergence after 2 iterations.
Gradient range [2.942202e-11,2.942202e-11]
(score 0.2114560 & scale 0.2043541).
Hessian positive definite, eigenvalue range
[0.001125860,0.001125860].

> anova(gam1b,gam1c,test="Chi")
Analysis of Deviance Table

Model 1: next7 ~ nint7 + I(nint7^2) + aext7 + aint7
Model 2: next7 ~ s(nint7) + aext7 + aint7
  Resid. Df Resid. Dev     Df Deviance P(>|Chi|)
1    180.00     37.558
2    178.79     36.536 1.2134   1.0219   0.03461 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> plot(gam1c,residuals=T,all.terms=T)
>
> gam1d <- gam(next7 ~ s(nint7)+aext7+I(aext7^2)+aint7,
+ family=Gamma(link="log"), data=redata2b)
> summary(gam1d)

Family: Gamma
Link function: log

Formula:
next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7

Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.1712362  0.0613870  35.370  < 2e-16 ***
aext7        0.0733160  0.0122305   5.995 1.11e-08 ***
I(aext7^2)  -0.0009745  0.0003437  -2.835  0.00511 **
aint7       -0.0229126  0.0073621  -3.112  0.00216 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
           edf Ref.df     F  p-value
```

```
s(nint7) 3.402   4.22 16.04 1.21e-11 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =   0.52   Deviance explained = 45.8%
GCV score = 0.20512  Scale est. = 0.19691   n = 185
> gam.check(gam1d)

Method: GCV   Optimizer: outer newton
full convergence after 2 iterations.
Gradient range [6.841015e-08,6.841015e-08]
(score 0.2051162 & scale 0.1969099).
Hessian positive definite, eigenvalue range [0.00088314,0.00088314].

> anova(gam1c,gam1d,test="Chi")
Analysis of Deviance Table

Model 1: next7 ~ s(nint7) + aext7 + aint7
Model 2: next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7
  Resid. Df Resid. Dev     Df Deviance P(>|Chi|)
1    178.79     36.536
2    177.60     34.971 1.1881   1.5649  0.006586 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> plot(gam1d,residuals=T,all.terms=T)
>
> gam1e <- gam(next7 ~ s(nint7)+s(aext7)+aint7,
+ family=Gamma(link="log"), data=redata2b)
> summary(gam1e)

Family: Gamma
Link function: log

Formula:
next7 ~ s(nint7) + s(aext7) + aint7

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.525929   0.058495  43.182  < 2e-16 ***
aint7       -0.022101   0.007357  -3.004  0.00305 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
           edf Ref.df     F  p-value
s(nint7) 3.351  4.157 16.26 1.16e-11 ***
s(aext7) 2.096  2.592 24.14 6.57e-12 ***
```

```
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.519   Deviance explained = 45.7%
GCV score = 0.20579  Scale est. = 0.1975    n = 185
> gam.check(gam1e)

Method: GCV   Optimizer: outer newton
full convergence after 3 iterations.
Gradient range [1.27168e-09,4.228609e-09]
(score 0.2057869 & scale 0.1975024).
Hessian positive definite, eigenvalue range
[0.0009633342,0.001436439].

> anova(gam1d,gam1e,test="Chi")
Analysis of Deviance Table

Model 1: next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7
Model 2: next7 ~ s(nint7) + s(aext7) + aint7
  Resid. Df Resid. Dev       Df  Deviance P(>|Chi|)
1    177.60     34.971
2    177.55     35.067 0.046102 -0.096131
> plot(gam1d,residuals=T,all.terms=T)
>
> gam1f <- gam(next7 ~ s(nint7)+aext7+I(aext7^2)+aint7+I(aint7^2),
+ family=Gamma(link="log"), data=redata2b)
> summary(gam1f)

Family: Gamma
Link function: log

Formula:
next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7 + I(aint7^2)

Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.1809030  0.0693531  31.446  < 2e-16 ***
aext7        0.0746719  0.0128693   5.802 2.97e-08 ***
I(aext7^2)  -0.0010095  0.0003591  -2.811  0.00549 **
aint7       -0.0274877  0.0161025  -1.707  0.08957 .
I(aint7^2)   0.0001814  0.0005647   0.321  0.74845
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
           edf Ref.df     F  p-value
s(nint7) 3.395  4.217 15.91 1.53e-11 ***
```

```
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.517   Deviance explained = 45.9%
GCV score = 0.20725  Scale est. = 0.19785   n = 185
> gam.check(gam1f)

Method: GCV   Optimizer: outer newton
full convergence after 2 iterations.
Gradient range [8.266135e-08,8.266135e-08]
(score 0.2072536 & scale 0.1978484).
Hessian positive definite, eigenvalue range
[0.000903898,0.000903898].

> anova(gam1d,gam1f,test="Chi")
Analysis of Deviance Table

Model 1: next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7
Model 2: next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7 + I(aint7^2)
  Resid. Df Resid. Dev      Df Deviance P(>|Chi|)
1    177.60     34.971
2    176.60     34.941 0.99377 0.029931    0.6948
> plot(gam1f,residuals=T,all.terms=T)
>
> gam1g <- gam(next7 ~ s(nint7)+aext7+I(aext7^2)+s(aint7),
+ family=Gamma(link="log"), data=redata2b)
> summary(gam1g)

Family: Gamma
Link function: log

Formula:
next7 ~ s(nint7) + aext7 + I(aext7^2) + s(aint7)

Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.0201366  0.0597070  33.834  < 2e-16 ***
aext7        0.0733160  0.0122306   5.994 1.11e-08 ***
I(aext7^2)  -0.0009745  0.0003437  -2.835  0.00511 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
           edf Ref.df      F  p-value
s(nint7) 3.402  4.220 16.044 1.21e-11 ***
s(aint7) 1.000  1.000  9.685  0.00216 **
---
```

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =   0.52   Deviance explained = 45.8%
GCV score = 0.20512  Scale est. = 0.19691   n = 185
> gam.check(gam1g)

Method: GCV   Optimizer: outer newton
full convergence after 9 iterations.
Gradient range [-1.046933e-07,4.042393e-09]
(score 0.2051163 & scale 0.1969099).
Hessian positive definite, eigenvalue range
[1.046854e-07,0.0008831097].

> anova(gam1d,gam1g,test="Chi")
Analysis of Deviance Table

Model 1: next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7
Model 2: next7 ~ s(nint7) + aext7 + I(aext7^2) + s(aint7)
  Resid. Df Resid. Dev         Df   Deviance P(>|Chi|)
1     177.6     34.971
2     177.6     34.971 0.00011785 2.8564e-05 0.0005275 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> plot(gam1g,residuals=T,all.terms=T)
>
> gam1h <- gam(next7 ~ s(nint7)+aext7+I(aext7^2)+aint7+
+ factor(nrchild), family=Gamma(link="log"), data=redata2b)
> summary(gam1h)

Family: Gamma
Link function: log

Formula:
next7 ~ s(nint7) + aext7 + I(aext7^2) + aint7
+ factor(nrchild)

Parametric coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      2.3486834  0.1051798  22.330  < 2e-16 ***
aext7            0.0737127  0.0121521   6.066 7.86e-09 ***
I(aext7^2)      -0.0009371  0.0003404  -2.753 0.006523 **
aint7           -0.0245764  0.0073351  -3.351 0.000988 ***
factor(nrchild)2 -0.2075395  0.0978887  -2.120 0.035399 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
```

```
           edf Ref.df     F  p-value
s(nint7) 3.601  4.458 16.29 3.22e-12 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =   0.52   Deviance explained = 46.6%
GCV score = 0.20111  Scale est. = 0.19171   n = 184
> gam.check(gam1h)

Method: GCV   Optimizer: outer newton
full convergence after 3 iterations.
Gradient range [1.049147e-09,1.049147e-09]
(score 0.2011097 & scale 0.1917089).
Hessian positive definite, eigenvalue range
[0.000900505,0.000900505].

> anova(gam1d,gam1h,test="Chi")
Error in anova.glmlist(c(list(object), dotargs),
 dispersion = dispersion,  : models were not
 all fitted to the same size of dataset
```