Introduction to GAMs and Quantile GAMs

Matteo Fasiolo

September 24, 2024

Contents

1 Introducing GAMs

1.1 GAM Model Structure

Consider a regression context, where y is a dependent variable with conditional distribution $p(y|\boldsymbol{x})$, $\boldsymbol{x}$ being a d-dimensional vector of covariates. In distributional regression, we are typically interested in modelling $p(y|\boldsymbol{x})$ via a parametric model, $p(y|\boldsymbol{\theta}, \boldsymbol{x})$, which is parametrized by the m-dimensional vector of parameters $\boldsymbol{\theta}$. The elements of $\boldsymbol{\theta}$ control various characteristic of the response distribution, such as location, scale and shape. In a standard regression modelling context, we allow only one of the elements of $\boldsymbol{\theta}$ to depend on $\boldsymbol{x}$. In the following, we call such parameter $\mu = \mu(\boldsymbol{x})$ and we use $\boldsymbol{\theta}$ to refer to the remaining parameters.

Parameter $\mu(\boldsymbol{x})$ is typically a location parameter, which controls the conditional mean of the response, $\mathbb{E}(y|\boldsymbol{x})$. For example, in a Gaussian regression model we assume that $y \sim \mathrm{N}\{\mu(\boldsymbol{x}), \sigma^2\}$ and parameter μ acts exclusively on the conditional mean, while the scale is controlled by σ. However, parameter μ might control multiple moments of the response distribution. In a Poisson regression model, for instance, we assume that $y \sim \mathrm{Poi}\{\mu(\boldsymbol{x})\}$, where $\mathbb{E}(y|\boldsymbol{x}) = \mathrm{var}(y|\boldsymbol{x})$, hence modelling the rate $\mu(\boldsymbol{x})$ results in both the mean and the variance being dependent on the covariates.

In GAM models, μ has a semi-parametric additive structure, that is

$$g\{\mu(\boldsymbol{x})\} = \boldsymbol{z}^{\mathsf{T}}\boldsymbol{\beta}^0 + \sum_{j=1}^{J} f_j(\boldsymbol{x}), \tag{1}$$

where g is a known monotonic function, $\boldsymbol{z} = \boldsymbol{z}(\boldsymbol{x})$ is d-dimensional vector whose value depends on the covariates $\boldsymbol{x}$ and the f_j's are smooth effects. Hence $\boldsymbol{z}^{\mathsf{T}}\boldsymbol{\beta}^0$ represents the parametric part of the model, with unknown regression coefficients $\boldsymbol{\beta}^0$. The f_j's are built using spline bases expansions, so the j-th effect can be written

$$f_j(\boldsymbol{x}) = \boldsymbol{b}_j^{\mathsf{T}}\boldsymbol{\beta}^j = \sum_{k=1}^{K_j} b_j^k(\boldsymbol{x})\beta_k^j,$$

where $\boldsymbol{b}_j = \{b_j^1, \ldots, b_j^{K_j}\}$ are the spline basis functions used to built the j-th effect and $\boldsymbol{\beta}^j = \{\beta_1^j, \ldots, \beta_{K_j}^j\}$ are the corresponding regression coefficients. The basis functions are known and fixed, while the regression coefficients must be estimated. The dependence of μ on $\boldsymbol{\beta}$ is linear, in fact we can write

$$g\{\mu(\boldsymbol{x})\} = \mathbf{x}^{\mathsf{T}}\boldsymbol{\beta},$$

where $\mathbf{x} = \{\boldsymbol{z}, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_J\}$ and $\boldsymbol{\beta} = \{\boldsymbol{\beta}^0, \boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^J\}$. Hence, the r.h.s. of (1) is generally called the "linear predictor". While μ depends on both $\boldsymbol{x}$ and $\boldsymbol{\beta}$, here we refer to μ using either $\mu(\boldsymbol{x})$ or $\mu(\boldsymbol{\beta})$, depending on context.

In `mgcv`, GAMs can be built and fitted using the `gam` function, an example call being

```
fit <- gam(formula = y ~ x1 + s(x2, k = 15, bs = "cr"),
           family = Poisson(link = log), data = SomeData)
```

The first argument is the model formula, where we are using a linear effect for covariate `x1` and a smooth effect for `x2`. Arguments `bs` and `k` of the smooth effect specifier, `s`, determine the type and number of basis functions used. See Section 1.3 for more details. The next argument of `gam` determines the response distribution to be used, here a Poisson distribution where the linear predictor is modelling $\log \mu(x_1, x_2) = \log \mathbb{E}(y|x_1, x_2)$. Under such model, one reason for using the log-link, $g = \log$, is to ensure the positivity of $\mu(x_1, x_2)$.

1.2 GAM Model Fitting in a Bayesian Framework

Consider a data set formed of responses $\mathbf{y} = \{y_1, \ldots, y_n\}$, covariate vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and let the log-likelihood of a GAM model for such data be $\mathrm{ll}(\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \log p\{y_i|\mu(\boldsymbol{x}_i), \boldsymbol{\theta}\}$. For the moment, assume that we are holding $\boldsymbol{\theta}$ fixed to some value, and that we want to estimate only the regression coefficients $\boldsymbol{\beta}$. Estimating $\boldsymbol{\beta}$ by maximizing $\mathrm{ll}(\boldsymbol{\beta}, \boldsymbol{\theta})$ is generally a bad idea in a GAM context, because it can result in severe overfitting, especially when the number of basis function used to build the smooth effects is large. This is because the likelihood quantifies how good is our fit on the observed data, but it does not penalise the complexity or wiggliness of the smooth effects. Hence, a fit based simply on maximizing the likelihood might overfit and not generalise well. The solution is estimating the regression coefficients by maximizing an alternative criterion, which contains a component aimed at controlling the wiggliness of the smooth effects, in addition to the likelihood.

Bayesian Statistics offers a principled approach for doing exactly this. In particular the complexity of the smooth effects is controlled using a prior distribution on the regression coefficients, which we indicate with $p(\boldsymbol{\beta}|\boldsymbol{\lambda})$. Here we assume that such prior is an improper multivariate Gaussian distribution, centred at $\mathbf{0}$ and with positive semi-definite precision matrix $\mathbf{S}^{\boldsymbol{\lambda}} = \sum_{l=1}^{m} \lambda_j \mathbf{S}_j$. The $\mathbf{S}_j$'s are positive semi-definite matrices, scaled by the positive parameters $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_m\}$. We assume that the $\mathbf{S}_j$'s are given, while the vector $\boldsymbol{\lambda}$ needs to be selected. As we will explain in Section 1.3, the structure of the $\mathbf{S}_j$'s is aimed at penalizing departures from (some definition of) smoothness of the corresponding effect $f_j(\boldsymbol{x})$. The result is that increasing $\boldsymbol{\lambda}$ leads to a prior on $\boldsymbol{\beta}$ under which the smooth effects are less wiggly. For this reason, in the following we refer to $\boldsymbol{\lambda}$ as the vector of smoothing parameters. In general, there needs not to be a one-to-one correspondence between the smoothing parameters and the effects because, for example, a smoothing parameter might be determining the prior precision of several effects, while the wiggliness of a single effect might be controlled using multiple smoothing parameters.

Under the Gaussian prior on $\boldsymbol{\beta}$, the log-posterior density is

$$\text{lp}(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \text{ll}(\boldsymbol{\beta}, \boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}^{\boldsymbol{\lambda}}\boldsymbol{\beta}, \tag{2}$$

up to an additive constant which does not depend on $\boldsymbol{\beta}$. For fixed $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$, maximizing the log-posterior w.r.t. leads to a maximum a posteriori (MAP) estimator, $\hat{\boldsymbol{\beta}}$. Given that (2) is maximized for fixed $\boldsymbol{\theta}$, the smoothing parameters should be selected so as to provide an adequate balance between fitting the data and keeping the effects smooth. `mgcv` offers several methods for doing this, but the most widely applicable is Laplace approximate marginal likelihood (LAML) smoothing parameter selection. The idea is to maximize a Laplace approximation to the marginal likelihood

$$\text{laml}(\boldsymbol{\lambda}, \boldsymbol{\theta}) \approx \log p(\mathbf{y}|\boldsymbol{\lambda}, \boldsymbol{\theta}) = \log \int p(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\theta}) p(\boldsymbol{\beta}|\boldsymbol{\lambda}) d\boldsymbol{\beta}, \tag{3}$$

w.r.t. $\boldsymbol{\lambda}$. It is possible to maximize (3) w.r.t. $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ jointly, thus providing a method for estimating the extra parameters of the response distribution. The alternative is to maximize (2) w.r.t. $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ jointly, which leads to the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$, because the implicit prior on $\boldsymbol{\theta}$ is flat. If the latter method is adopted, the LAML becomes a function of $\boldsymbol{\lambda}$ only, because both $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ must be integrated out using a Laplace approximation. It is important to point out that LAML is equivalent to the restricted maximum likelihood (REML) criterion, as defined in Laird et al. (1982). An alternative to LAML is maximum likelihood (ML) estimation, where the integral (3) is (approximately) perfomed only across the range space of the prior precision (i.e. its null space is excluded). See Wood et al. (2016) for more details on LAML smoothing parameter selection in general GAM models and Wood (2011) for a brief discussion of alternative methods.

1.3 Basic Smooth Effects and Penalties

The `mgcv` `R` package provides a wide variety of smooth effects types, which are formed by a basis vector and one or more penalties. Here we focus on derivative-based penalties, because they are useful in many applications, but we refer to Wood (2017) for a comprehensive overview.

1.3.1 Thin Plate Splines and Derivative-based Penalties

Consider a univariate smooth effect, $f(x)$, formed by a vector of basis functions, $\mathbf{b}(x)$. Let $\boldsymbol{\beta}$ be the corresponding vector of regression coefficients, on which we impose a Gaussian prior with precision matrix $\mathbf{S}$. Now, the purpose of the prior is limiting the wiggliness of $f(x)$ and the prior's strength is determined by the smoothing parameter, λ. To see how a prior precision matrix can be related to a penalty on wiggliness, let $f^{(j)}$ be the j-th derivative of f and consider the penalty $q_{j1}(f) = \int f^{(j)}(x)^2 dx$. We have that

$$q_{j1}(f) = \int f^{(j)}(x)^2 dx = \int \{\mathbf{b}^{(j)}(x)^\mathsf{T}\boldsymbol{\beta}\}^2 dx \tag{4}$$

$$= \boldsymbol{\beta}^\mathsf{T}\left\{\int \mathbf{b}^{(j)}(x)\mathbf{b}^{(j)}(x)^\mathsf{T} dx\right\}\boldsymbol{\beta} = \boldsymbol{\beta}^\mathsf{T}\mathbf{S}\boldsymbol{\beta}, \tag{5}$$

where we used $\mathbf{b}^{(j)}(x)^\mathsf{T}\boldsymbol{\beta} = \boldsymbol{\beta}^\mathsf{T}\mathbf{b}^{(j)}(x)$ and defined $\mathbf{S} = \int \mathbf{b}^{(j)}(x)\mathbf{b}^{(j)}(x)^\mathsf{T} dx$. Hence, under such definition of $\mathbf{S}$, maximizing the log-posterior is equivalent to maximizing the penalized log-likelihood, $ll(\boldsymbol{\beta}, \boldsymbol{\theta}) - \lambda\, q_{j1}(f)/2$. $q_{j1}(f)$ is a derivative-based penalty, which provides a simple example of how a smoothness penalty can induce a prior precision structure. In this example the prior is shrinking the smooth effect toward a straight line.

So far the basis vector $\mathbf{b}(x)$ was given, now we examine the question of what classes of basis functions should be considered in a regression smoothing context. We do this in the more general case of a multidimensional effect $f(\boldsymbol{x})$, with $\boldsymbol{x} \in \mathbb{R}^d$. In particular, consider the problem of finding the function f minimizing

$$\sum_{i=1}^{n}\{y_i - f(\boldsymbol{x}_i)\}^2 + \lambda q_{md}(f), \tag{6}$$

where

$$q_{md}(f) = \int_{\mathbb{R}^d} \sum_{\boldsymbol{j}\in J_d} \frac{m!}{j_1! \cdots j_d!} f^{(m)}_{j_1 \cdots j_d}(\boldsymbol{x})^2 d\boldsymbol{x}. \tag{7}$$

Here J_d indicates the set of all vectors $\boldsymbol{j} = \{j_1, \ldots, j_d\}$ such that $\sum_{k=1}^{d} j_k = m$ and $j_k \in \{0, 1, \ldots, m\}$, while $f^{(m)}_{j_1 \cdots j_d}$ is the m-th derivative of f, obtained by differentiating it j_1 times w.r.t. x_1, j_2 times w.r.t. x_2 and so on. Expression (7) might look intimidating but, if we penalize the second derivative, we recover $q_{21}(f)$ in one dimension, while in two dimensions we have $q_{22}(f) = \int f^{(2)}_{20}(\boldsymbol{x})^2 + 2f^{(2)}_{11}(\boldsymbol{x})^2 + f^{(2)}_{02}(\boldsymbol{x})^2 d\boldsymbol{x}$.

It turns out that, if $2m > d$, an optimal solution to the minimization of (6) is provided by thin plate splines, for short t.p.s. (Duchon, 1977). In particular, t.p.s. are smooth functions that can be expressed in terms of a basis formed by a set of polynomial functions and a set of more complex functions, each of the latter being centred at one of the n observed values of $\boldsymbol{x}$. The exact form of the basis is unimportant here, what matters is that it leads to an optimal solution to (6) and that the number of unknown free parameters is n. The number of free parameters is also called the rank of the effect, and it is obtained by subtracting the number of identifiability constraints from the number of basis coefficients, which is larger than n for t.p.s.. It is remarkable that t.p.s. provide an optimal solution to such a general and practically relevant minimization problem, but having as many parameters as data points is problematic from a computational point of view, hence in practice it is often preferable to reduce the number of basis function, and corresponding coefficients, to $K \ll n$. Obviously,

reducing the number of basis functions will compromise the optimality of the t.p.s. solution, and the question is how to choose a basis of size K which leads to a function f that is closest, in some sense, to the full t.p.s. solution.

Thin plate regression splines, for short t.p.r.s., (Wood, 2003) provide one answer to this question, by adopting the basis of size K leading to the fitted function f which simultaneously minimizes the worst possible changes in fitted value and in shape, relative to the full t.p.s. solution. Hence, t.p.r.s. can be seen a least-worst (or minimax) approximation to t.p.s., and belong to the wider class of eigen-based approximations, because the rank of the smoother is reduced using a truncated eigen-decomposition. Eigen-based dimension reduction methods are applicable also to non-derivative-based smoothers, such as Markov random fields and Gaussian processes, but these will not be covered here, hence we refer the interested reader to Chapter 5 of Wood (2017).

A different class of low-rank spline approximations is obtained using knot-based methods. These can be motivated by that fact that, when $m = 2$ and $d = 1$, an alternative representation of the unique function f minimizing (6) is a cubic spline, which is formed by a sequence third-order polynomials P_k such that $f(x) = P_k(x)$ when $x \in [x_k, x_{k+1}]$, where x_k is the k-th ordered value of x with $k \in \{1, \ldots, n-1\}$. It must be stressed that, under the $q_{21}(f)$ penalty, cubic splines and t.p.s. offer different basis representations of the same optimal smooth function f. The polynomials are subject to continuity and smoothness constraints, which make so that the number of unknown free parameters is n, thus leading to the same computational issue as t.p.s.. Cubic regression splines (c.r.s.) offer a knot-based solution to the problem, and are obtained by using $K - 1 \ll n$ polynomials defined on $[x_1^0, x_2^0], \ldots, [x_{K-1}^0, x_K^0]$. The x_k^0's are called knots, and the resulting approximation to the full spline has K unknown parameters. Knot-based approximations can be applied more widely, for example to t.p.s. and Gaussian processes. Under such smoothers, the knots often represent the locations in $\mathbb{R}^d$ at which the basis functions are centered, in some sense. Reducing the rank of the smoother using K, rather than n, knots is computationally cheap (e.g., no eigen-decomposition is required), but optimally choosing the location of the knots is a non-trivial problem and simple solutions, such as evenly spacing the knots, often struggle when $d > 1$. Hence, eigen-based rank reduction methods often lead to better approximations to the full rank spline.

1.3.2 Smooth Effects in `mgcv`

In `mgcv`, smooth effects can be specified by using the `s()` function inside the model formula supplied to a call to the `gam` function. For example, `s(x, bs = "tp", k = 10, m = 3)` sets up a one-dimensional smooth effect for covariate x, built using a t.p.r.s. basis functions vector $\mathbf{b}(x)$ of dimension 10, and the corresponding $q_{31}(f)$ penalty. This smoother naturally generalizes to higher dimensions hence, for instance, `s(x1, x2, bs = "tp", k = 25)` sets up a 2D t.p.r.s. basis of dimension 25, with penalty $q_{22}(f)$. A c.r.s. knot-based smoothers is specified using `s(x, bs = "cr")`. A variant of c.r.s. that is often useful in applications is a cyclical c.r.s., obtained by setting `bs = "cc"`. Cyclical c.r.s. smoothers have the property that $f^{(i)}(x_{\min}) = f^{(i)}(x_{\max})$ for $i = 0, 1$ and 2, where $f^{(i)}$ is the i-th derivative of f, while $x_{\min}$ and $x_{\max}$ are the smallest and largest observed values of x, unless the locations of the extremal knots are supplied manually.

As explained before, a t.p.r.s. smoother is an approximation to a full t.p.s. spline basis, the latter being the minimizer of (6), when $2m > d$. When $d = 1$ and $m = 2$, c.r.s. provide a different approximation to the same optimal t.p.s. function. Hence, given d and the number

of basis or knots positions, specifing the order m of the penalty implies that a certain t.p.r.s. or c.r.s. basis is used. This mean that the relation between the penalty and the basis is fixed. For example, it is not possible (in `mgcv 1.8.30` or earlier version, at least) to use a c.r.s. basis with a $q_{11}(f)$ penalty, or to construct a t.p.r.s. approximation to the full t.p.s. minimizing (6) for $q_{32}(f)$, but to fit it using a $q_{22}(f)$ penalty. B-splines are attractive in this sense because, while spanning the **same space of functions** as standard polynomial splines, they allow mixing spline bases and penalties of different orders. Such smoothers are specified using, e.g., `s(x, bs = "bs", m = c(3, 1, 0))`, where the first entry of the vector `m` indicates that we are using a B-spline basis representing a third-order (cubic) spline, while the remaining entries add a $q_{11}(f)$ and a $q_{01}(f)$ penalties. Hence, B-splines can be associated with multiple penalties, each scaled by a different smoothing parameter.

P-splines (Eilers and Marx, 1996) are smoothers based on B-splines bases, with one or more penalties that do not directly penalize the derivatives of f, but the differencies between regression coefficients. In particular, a P-spline penalty of order one is $\tilde{q}_{11}(f) = \sum_{k=1}^{K-1}(\beta_{k+1} - \beta_k)^2$, while a second order penalty is $\tilde{q}_{21}(f) = \sum_{k=2}^{K-1}(\beta_{k+1} - 2\beta_k + \beta_{k-1})^2$. P-splines are specified using, e.g., `s(x, bs = "ps", m = c(2, 2, 1))`, where we are using a third order B-spline basis[1] with second and first order P-spline penalties on $\boldsymbol{\beta}$. P-spline penalties are particularly useful for setting up adaptive smoothers, where the strength of the wiggliness penalty is allowed to vary along the covariate of interest, x. More precisely, under such smoothers, the smoothing parameter of the P-spline penalty is itself modelled using B-splines, so that $\lambda = \lambda(x)$. As detailed in section 5.3.5 of Wood (2017), such adaptive smoothers can be expressed as standard spline effects, constructed using cubic or B-spline bases, with multiple P-spline penalties. Each of the latter has its own smoothing parameter and penalizes the wigglines of $f(x)$ on a finite interval along x, but it partially overlaps with penalties affecting adiacent sections of x. Adaptive smoother are specified by, e.g., `s(x, bs = "ad", k = 40, m = 4, xt = list(bs = "cr"))`, where we are using an adaptive smoother, built using a c.r.s. basis with 40 knots and four smoothing parameters. Adaptive effects can be useful in certain situations, but they must be used with parsimony, as the presence of multiple smoothing parameters makes them particularly data-hungry.

We conclude this section by pointing out that, before fitting a GAM model, the number of basis functions used to build a smooth effect must be reduced, to avoid identifiability problems. For example, most spline bases $\mathbf{b}(x)$ contain the constant function in their span, that is, for any $c \in \mathbb{R}$, there exists a vector $\boldsymbol{\beta}$ such that $\boldsymbol{\beta}^\mathsf{T}\mathbf{b}(x) = c$. It is easy to see that a model containing two or more such effects is not identifiable. To deal with this issue, `mgcv` imposes a sum-to-zero constraint on most effects, so that $\sum_{i=1}^{n} f(x_i) = 0$ where $x_1, \ldots, x_n$ are the observed values of x. The constraint does not affect the value of derivative-based or P-spline penalties of order greater than zero, because it does not perturb the shape of $f(x)$, but simply shifts it vertically. Given a basis of size K, `mgcv` imposes the constraint by removing the constant function from the span of the basis, thus reducing its rank by one. `mgcv` can deal with more complex constraints, which are often needed to address the identifiability issues implied by the specific combination of effects used in the model formula supplied to `gam`. In summary if, in a call to `gam`, we specify that an effect should use a basis of rank K, but the output of `gam` shows that only $K' < K$ basis functions were used, the loss in basis rank is

[1] `mgcv`'s order convention can be confusing. In the smoothing splines literature, a B-spline of order m corresponds to a polynomial spline with power $m + 1$. So, setting `m[1] = 2` when `bs = "ps"`, creates a third order B-spline basis, which is a knot-based approximation to a full cubic spline. But, when `bs = "bs"`, constructing the same basis requires setting `m[1] = 3`.

generally attributable to identifiability constraints.

1.4 Model Selection

Model selection is an important task in applied Statistics, hence here we briefly discuss how to perform it in a GAM context.

1.4.1 Model Selection via Smoothing Parameter Estimation

The first thing to point out is that, while in parametric regression models the effect of a covariate x_j is either included in the model or omitted, penalized regression models define a continuum of nested models. In particular, the estimated value of the smoothing parameters will determine where the fitted GAM model lies, in the space of models ranging from completely unpenalized ones ($\boldsymbol{\lambda} = \mathbf{0}$) to those where the estimated coefficients are forced to lie in the null-space of the penalty ($\boldsymbol{\lambda} = \infty$). For example, the penalty matrix $\lambda\mathbf{S}$ induced by a derivative-based $q_{21}(f)$ penalty has a null space of dimension two, because it does not effect straight lines, which is reduced to one after the intercept is removed from the basis to satisfy a sum-to-zero constraint. This means that, if $\lambda \to \infty$, a smooth effect controlled by such a penalty has only one free parameter (the slope). Therefore, a good part of GAM model selection is performed automatically via smoothing parameter selection.

Standard GAM smoothing parameter selection methods can be used to completely remove an effect, that is to shrink it to a flat horizontal line, rather than to a function in the null space of the penalty. For example, if the smooth effect is continuous, full penalization can be achieved by adopting a $q_{11}(f)$ penalty. Under a piecewise constant effect built using, e.g., a B-spline basis of order zero, full penalization is achieved using either a $\tilde{q}_{11}(f)$, $\tilde{q}_{01}(f)$ or $q_{01}(f)$ penalty. The `gam` function in `mgcv` provides a `select` argument which, if set to `TRUE`, creates extra penalties affecting the null spaces of each of the smooth effects contained in the model. Null space penalization makes so that all the smooth effects corresponding to smoothing parameters that have diverged ($\boldsymbol{\lambda} \to \infty$) will be completely flat. In practice, smoothing parameter selection generally stops short of infinity, leading to "almost flat" effects, which must be removed manually from the model. For this reason, methods based on null space penalties are less than ideal when a hard binary model selection decision (i.e., to include or to exclude a term) is desired. Such questions are better addressed using model selection criteria, which are useful also when the models being compared are not nested.

1.4.2 Performing AIC-based Model Selection under Penalization

Akaike's information criterion (AIC, Akaike, 1973) is one of the most widely used statistical model selection criteria. The idea is to choose the model minimizing

$$\text{AIC} = -2\text{ll}(\hat{\boldsymbol{\psi}}) + 2\tau,$$

where $\text{ll}(\hat{\boldsymbol{\psi}})$ is the log-likelihood of a model with parameter vector $\boldsymbol{\psi}$, evaluated at its maximizer, and τ is the number of parameters, $\dim(\boldsymbol{\psi})$. In a GAM context, $\hat{\boldsymbol{\psi}} = \{\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}\}$, where $\hat{\boldsymbol{\beta}}$ is a MAP estimator and $\hat{\boldsymbol{\theta}}$ is either an ML or a LAML estimator, but defining τ is more problematic. For example, a c.r.s. effect with K knots can effectively have only one free parameter (the slope), if the smoothing parameter of its $q_{21}(f)$ penalty is very positive. Hence, while the nominal number of parameters is K, the number of effective degrees of freedom (e.d.f.)

is one. Thus, making AIC work for penalized models, requires setting τ to an appropriate estimate of the e.d.f. used by the fitted model. In particular, if $\boldsymbol{\theta}$ is estimated using LAML, the e.d.f. can be approximated by

$$\tau = \text{tr}\left\{\text{cov}(\hat{\boldsymbol{\beta}})\hat{\mathcal{I}}\right\} + \dim(\boldsymbol{\theta}), \tag{8}$$

where tr indicates the trace operator, $\hat{\mathcal{I}}$ is the negative Hessian of the log-likelihood (i.e., the observed Fisher information matrix) and $\text{cov}(\hat{\boldsymbol{\beta}})$ is the covariance matrix of the MAP estimator. Here we do not delve into the derivation of (8) but we point out that, if there is no penalization ($\boldsymbol{\lambda} = \mathbf{0}$) and the model is well-specified, $\text{cov}(\hat{\boldsymbol{\beta}})$ can be estimated by $\hat{\mathcal{I}}^{-1}$ and τ becomes equal to $\dim(\{\boldsymbol{\beta}, \boldsymbol{\theta}\})$. Instead, as $\boldsymbol{\lambda} \to \infty$, τ converges to the sum of $\dim(\boldsymbol{\theta})$ and of the dimension of the null space of $\mathbf{S}^{\boldsymbol{\lambda}}$. For intermediate values of $\boldsymbol{\lambda}$, τ will fall between these extremes and it is obtained by estimating $\text{cov}(\hat{\boldsymbol{\beta}})$ using the asymptotic Bayesian posterior covariance $\mathbf{V}_{\boldsymbol{\beta}}$, derived under a Gaussian approximation to the posterior. The performance of the resulting AIC is improved if $\mathbf{V}_{\boldsymbol{\beta}}$ is substituted by a corrected version, $\mathbf{V}'_{\boldsymbol{\beta}}$, which takes into account the uncertainty in the smoothing parameter estimates. `mgcv` uses the corrected version whenever possible. If the unpenalized extra parameters $\boldsymbol{\theta}$ are estimated by ML, the only difference is that $\hat{\mathcal{I}}$ in (8) now indicates the submatrix of the full negative Hessian containing the (mixed) second derivatives in $\boldsymbol{\beta}$ only. For more details, see section 6.11 of Wood (2017), where this is referred to as the "conditional" AIC.

1.4.3 Choosing the Type and the Basis Dimension of a Smooth Effect

In Section 1.3 we described some of the smooth effect types provided by `mgcv`. Given the array of available effects, a practitioner might wonder whether there are some general principles that can be used to select what type of smoother should be used for a particular effect. In general, spline bases built using eigen-based rank reduction methods are preferable to knot-based alternatives, because they explicitly minimize an upper bound on the approximation error caused by the rank reduction. Manual knot positioning can occasionally improve the quality of knot-based approximations, but it is time consuming. A frequently observed scenario where such an approach disappoints occurs when the shape of, say, a c.r.s. fit suggests that the wiggliness of the effect $f(x)$ should vary with x. In such cases, placing more knots where we would like the effect to be more flexible is often not a solution, as the associated $q_{21}(f)$ penalty penalizes wiggliness uniformly across x. A better solution is transforming the covariate x, with the aim of "stretching" the intervals of x-axis where we would like the effect to be more flexible, or using an adaptive smoother. In general the choice of effects needed should be determined by case specific considerations regarding the nature of the covariate(s) along which the smooth is defined, rather than by an automated procedures aiming at minimizing the forecasting error on a test set. However, such automated prediction-oriented methods are occasionally useful, for example when modelling data consisting of a large number of heterogeneous time series.

Domain specific knowledge about the application of interest can be integrated with, and potentially invalidated by, model-based diagnostics. A number of standard diagnostics are provided by `mgcv`, in particular methods for checking whether the number of basis functions used for a smooth effect is sufficiently large. Checking this is important, because the number of basis functions used for an effect put a upper bound on its flexibility, and **the default basis dimensions used by `mgcv` are arbitrary**. Using a reasonably generous number of

basis functions is generally not a problem, because smoothing parameter selection via LAML should avoid overfitting by reducing the number of e.d.f. used for an effect.

2 Introducing QGAM Models

Let y be the response or dependent variable with conditional distribution $p(y|\boldsymbol{x})$, where $\boldsymbol{x}$ is a d-dimensional vector of covariates. As explained in the previous Section, under a GAM model $p(y|\boldsymbol{x})$ is approximated by the parametric distribution $p(y|\boldsymbol{\theta}, \boldsymbol{x})$, where $\boldsymbol{\theta}$ is an m-dimensional vector of model parameters. In a GAM model only one element of $\boldsymbol{\theta}$ is allowed to vary with the covariates, via an additive model which can include several types of parametric and smooth effects. The value of the other parameters are generally unknown and must be estimated, but do not vary with $\boldsymbol{x}$. In most cases of practical interest, the parameter being modelled additively is a location parameter, controlling $\mathbb{E}(y|\boldsymbol{x})$. The other moments (e.g., variance or skewness) of the response distribution either do not vary with $\boldsymbol{x}$ or depend directly on the location parameter via a distribution-specific relationships (e.g., in a Poisson model $\text{var}(y|\boldsymbol{x}) = \mathbb{E}(y|\boldsymbol{x})$).

In the electricity industry, there are many practical contexts where it is important to move beyond mean modelling. For example, a short-term estimate of the expected electricity demand is not necessarily the most relevant quantity for trading purposes. In fact, $\mathbb{E}(y|\boldsymbol{x})$ is the minimiser of the quadratic loss, which is generally different from the economic loss that must be minimised when developing a trading strategy. Better results can be achieved by estimating the full distribution of $y|\boldsymbol{x}$, using it to estimate $\mathbb{E}\{L(y, D)|\boldsymbol{x}\}$, where L is the loss of interest, and minimising it w.r.t. the decision D. Similar considerations hold for production planning problems, where the aim is not simply matching production with the expected demand, but minimising a more complex loss which takes into account the cost of over-production, of buying electricity on the spot market and the risk of disruptions to customers, in addition to many other factors. Further, full probabilistic modelling becomes more important when forecasting electricity demand at low levels of aggregation. In fact, the signal-to-noise ratio decreases as the demand is averaged across fewer customers, so modelling the demand variability becomes more important than providing point estimates. Developing an adequate model for $p(y|\boldsymbol{x})$ can be important even when $\mathbb{E}(y|\boldsymbol{x})$ is the only quantity of interest. In fact, the Bayesian GAM fitting framework outlined in the previous Section, implicitly relies on the assumption that $p(y|\boldsymbol{\theta}, \boldsymbol{x})$ is a good approximation to $p(y|\boldsymbol{x})$, for some value of $\boldsymbol{\theta}$. In particular, automatic smoothing parameters selection via marginal likelihood methods can fail to provide adequate levels of smoothness when the response distribution is misspecified. An extreme example of this issue is discussed by Fasiolo et al. (2020a), who show that fitting quantile GAM models, which can be seen as misspecified GAMs, requires a modified fitting framework.

2.1 QGAM Model Structure

When the full conditional response distribution is of interest, it is common practice to summarise the predicted distribution using a set of conditional quantiles. Under probabilistic GAM models, these are obtained by inverting (an estimate of) the model-based conditional c.d.f. $F(q|\boldsymbol{\theta}, \boldsymbol{x}) = \text{Prob}(y \leq q|\boldsymbol{\theta}, \boldsymbol{x})$. More precisely

$$q_\tau(\boldsymbol{\theta}, \boldsymbol{x}) = \inf\{y : F(y|\boldsymbol{\theta}, \boldsymbol{x}) \geq \tau\} = F^{-1}(\tau|\boldsymbol{\theta}, \boldsymbol{x}),$$

where $\tau \in [0,1]$ and the second equality holds when y is a continuous variable. Hence, the quantile corresponding to any probability level τ can be obtained easily from the c.d.f. corresponding to the chosen parametric distribution.

Quantile regression models (Koenker, 2005) avoid any parametric distributional assumption about $p(y|\boldsymbol{x})$ by modelling and estimating each conditional quantile individually. This is achieved by exploiting the following alternative definition of a conditional quantile

$$q_\tau(\boldsymbol{x}) = \underset{q}{\operatorname{argmin}}\, \mathbb{E}\{\rho_\tau(y-q)|\boldsymbol{x}\}, \tag{9}$$

where

$$\rho_\tau(z) = (\tau-1)\frac{z}{\sigma}\mathbb{1}(z<0) + \tau\frac{z}{\sigma}\mathbb{1}(z\geq 0)$$

is the scaled version of the so-called 'pinball' or 'check' loss. The role of the scale parameter $\sigma > 0$ will be clarified in Section 2.2. As for distributional GAMs, we assume that the conditional quantiles have an additive structure, that is

$$q_\tau(\boldsymbol{x}) = \boldsymbol{z}^\mathsf{T}\boldsymbol{\beta}^0 + \sum_{j=1}^{J} f_j(\boldsymbol{x}).$$

As for standard GAMs, we control the wiggliness of the corresponding quantile GAM (QGAM) model (Fasiolo et al., 2020a) via a Gaussian smoothing prior with precision matrix $\mathbf{S}^{\boldsymbol{\lambda}}$, where $\boldsymbol{\lambda}$ is a vector of positive smoothing parameters.

In this chapter we focus on QGAM modelling via the `qgam` `R` package (Fasiolo et al., 2020b), which is an extension of `mgcv`. QGAM models can be built and fitted to data using `mgcv`-like code such as

```
fit <- qgam(formula = y ~ x1 + s(x2, k = 5), data = SomeData, qu = 0.9)
```

which fits the QGAM model $q_\tau(\boldsymbol{x}) = x_1 + f(x_2)$ to estimate quantile $\tau = 0.9$ of the response y. Even though the structure of QGAM models reseambles that of GAMs, the former are based on the pinball loss rather than on a parametric model for the response distribution. The fact that QGAM model fitting is loss-based, rather than likelihood-based, requires the modification to standard GAM model fitting methods outlined in Section 2.2.

2.2 Fitting QGAM Models with `qgam`

Here we briefly outline the Bayesian QGAM fitting methods proposed by (Fasiolo et al., 2020a) and implemented in the `qgam` package. The first difficulty that must be faced when fitting QGAMs in a Bayesian framework, is that the lack of the likelihood $p(\mathbf{y}|\boldsymbol{\beta},\sigma) = \prod_{i=1}^n p(y_i|\boldsymbol{\beta},\sigma)$ impedes the application of Bayes rule to update the Gaussian smoothing prior $p(\boldsymbol{\beta}|\boldsymbol{\lambda})$ to the corresponding posterior $p(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{\lambda},\sigma) \propto p(\mathbf{y}|\boldsymbol{\beta},\sigma)p(\boldsymbol{\beta}|\boldsymbol{\lambda})$. Following the coherent Bayesian updating framework of Bissiri et al. (2016), `qgam` adopts the following loss-based updating rule

$$p(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{\lambda},\sigma) \propto \exp\left\{-\sum_{i=1}^{n}\rho_\tau\left(\frac{y_i - q_\tau(\boldsymbol{x}_i)}{\sigma}\right)\right\} p(\boldsymbol{\beta}|\boldsymbol{\lambda}). \tag{10}$$

where $\exp\{q_\tau((y-q)/\sigma)\}$ is, up to a normalising constant, the p.d.f. of an asymmetric Laplace (AL) distribution with location q and scale σ. Hence, taking the exponential of the negative

loss allows us to form a pseudo-likelihood, based on the AL density, which can be used to perform the Bayesian update. See Bissiri et al. (2016) for details on the coherency properties of this update rule, which is also known at the Gibbs posterior and applies to losses other than the pinball loss.

Define $\text{lp}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \sigma) = \log p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{\lambda}, \sigma)$ and recall that, under standard GAMs, we obtain a fast maximum a posteriori (MAP) estimate of $\boldsymbol{\beta}$ by maximising $\text{lp}(\boldsymbol{\beta}, \boldsymbol{\lambda}, \sigma)$ w.r.t. $\boldsymbol{\beta}$, while keeping $\boldsymbol{\lambda}$ and σ fixed. Under a GAM model, $\boldsymbol{\lambda}$ and σ are selected by maximising a Laplace approximation to the marginal likelihood (LAML) (see the first section of this document for details and note that therein σ is substituted by a vector of parameters $\boldsymbol{\theta}$). Both the inner (MAP) and the outer (LAML) optimisation can be performed using efficient numerical methods, provided that the log-likelihood has several continuous derivatives. This is not the case under the AL-based pseudo-log-likelihood, which is piecewise linear. Fasiolo et al. (2020a) address the issue by substituting the pinball loss with a smoother extended log-F (ELF) loss

$$\tilde{\rho}_\tau(z) = (\tau - 1)\frac{z}{\sigma} + \psi \log(1 + e^{\frac{z}{\psi\sigma}}). \tag{11}$$

The smoothness of the ELF loss is controlled by $\psi > 0$, so that the pinball loss is recovered for $\psi \to 0^+$. In `qgam`, parameter ψ is selected via a preliminary step (that is, before estimating $\boldsymbol{\beta}$, $\boldsymbol{\lambda}$ and σ) by minimising an estimate of the asymptotic MSE of $\hat{\boldsymbol{\beta}}_{ML}$, the maximum likelihood estimator of $\boldsymbol{\beta}$.

Having determined the value of ψ, the smooth ELF loss can be used in place of the pinball-loss in (10), thus enabling the use of efficient MAP and LAML methods to estimate the remaining model parameters. However, selecting the value of σ via LAML maximisation leads to poor results, especially when fitting extreme quantiles ($\tau \approx 0$ or $\tau \approx 1$). The issue, which is explained in detail in Fasiolo et al. (2020a), is that the ELF-based response model is highly misspecified by construction. In particular, while the use of $\exp\{\tilde{\rho}_\tau\}$ as a pseudo-likelihood is justified by the framework of Bissiri et al. (2016), the ELF density does not appropriately describe the distribution of the response y. Hence, when selecting σ, it is useful not to think of it as the scale parameter of the ELF distribution, to be estimated via LAML as the scale parameter of a standard GAMs, but as a parameter controlling the relative weight of the ELF pseudo-likelihood and of the prior in the Gibbs posterior. In particular, looking at (10), it is clear that the AL pseudo-likelihood will increasingly dominate the prior for $\sigma \to 0^+$, while increasing σ has the opposite effect. Parameter σ plays a similar role when the ELF loss is adopted, hence we can see $\nu = 1/\sigma$ as a "learning rate" parameter, controlling the speed at which the prior is updated via the pseudo-likelihood. While such a parameter could be selected in several ways (see Bissiri et al. (2016) for some discussion), `qgam` implements the calibration methods proposed by Fasiolo et al. (2020a).

The idea is to choose σ such that the posterior credible intervals of the estimated quantile $q_\tau(\boldsymbol{x})$ have approximately the correct frequentist coverage. More precisely, let $C_\alpha\{\sigma, \mathbf{y}\}$ be the credible interval for $q_\tau(\boldsymbol{x})$, at probability level $\alpha \in (0, 1)$ (e.g., $\alpha = 0.95$ for 95% intervals). The aim is selecting σ so that

$$\mathbb{P}\big[q^0_\tau(\boldsymbol{x}) \in C_\alpha\{\sigma, \mathbf{y})\}\big] \approx \alpha, \tag{12}$$

for all α, where $\mathbb{P}$ is the objective probability measure, based on the data-generating process, and $q^0_\tau(\boldsymbol{x})$ is the true conditional quantile. In plain words, σ is chosen so that $\alpha 100\%$ intervals contain the true quantile approximately $\alpha 100\%$ of the time. This is achieved by numerically minising an integrated KullbackLeibler (IKL) divergence criterion w.r.t. σ. The details are

beyond the scope of this document, but see Fasiolo et al. (2020a) if you would like to know more.

In summary, QGAM model fitting with `qgam` entails the following steps:

1. select the loss smoothness parameter ψ via a preliminary step. The chosen value of ψ will be held fixed throughout the rest of the fitting procedure.

2. Select σ by numerical IKL minimization. To evaluate the IKL at a trial value of σ, we need to estimate $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$. For fixed σ, this requires:

   - selecting $\boldsymbol{\lambda}$ via LAML maximisation. To evaluate the LAML criterion at a trial value of $\boldsymbol{\lambda}$, we need to estimate the regression coefficients. For fixed σ and λ, this requires:
     - numerically maximising the ELF-based log-posterior w.r.t. $\boldsymbol{\beta}$.

Hence, step 2 of the procedure above can be thought of as three nested `for` loops: the outer one selects σ, the intermediate one selects $\boldsymbol{\lambda}$ for fixed σ and the inner one estimates $\boldsymbol{\beta}$ for fixed σ and $\boldsymbol{\lambda}$. In a standard GAM, parameter σ would be estimated jointly with $\boldsymbol{\lambda}$ via LAML estimation, thus avoiding the outer loop. Under the current version 1.3.2 of `qgam`, the need to explicitly calibrate the σ using an outer IKL minimisation procedure makes so that fitting a QGAM model with `qgam` is much slower than fitting a GAM with a similar model structure with `mgcv`. How to avoid this is the focus of current research.

The fitting framework just described can be extended to allow the learning rate to depend on the covariates, that is $\sigma = \sigma(\boldsymbol{x})$. This can be achieved by estimating the conditional variance via a parametric (eg., Gaussian) GAM and using it to form $\sigma(\boldsymbol{x}) = \sigma_0\kappa(\boldsymbol{x})$, where σ_0 is selected by IKL calibration as explained above and $\kappa(\boldsymbol{x})$ is an estimate of $\text{var}(y|\boldsymbol{x})$. Making the speed of learning inversely proportional to the conditional variance make so that the model will learn more slowly were the response variable is more noisy, hence less informative, which a desirable property. Also, for fixed ψ, the ELF loss will be smoother in areas of the covariates space where $\text{var}(y|\boldsymbol{x})$ is larger, which also advantageous (see Fasiolo et al. (2020a) for details).

References

Akaike, H. (1973). Information theory as an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory. Akademiai Kiado, Budapest*, pp. 267–281.

Bissiri, P., C. Holmes, and S. Walker (2016). A General Framework for Updating Belief Distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 78*(5), 1103–1130.

Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in sobolev spaces. In W. Schempp and K. Zeller (Eds.), *Constructive Theory of Functions of Several Variables*, Berlin, Heidelberg, pp. 85–100. Springer Berlin Heidelberg.

Eilers, P. H. and B. D. Marx (1996). Flexible Smoothing with B-splines and Penalties. *Statistical Science 11*(2), 89–102.

Fasiolo, M., S. N. Wood, M. Zaffran, R. Nedellec, and Y. Goude (2020a). Fast calibrated additive quantile regression. *Journal of the American Statistical Association 0*(0), 1–11.

Fasiolo, M., S. N. Wood, M. Zaffran, R. Nedellec, and Y. Goude (2020b). qgam: Bayesian non-parametric quantile regression modelling in r. *arXiv preprint arXiv:2007.03303*.

Koenker, R. (2005). *Quantile regression*. Cambridge: Cambridge University Press.

Laird, N. M., J. H. Ware, et al. (1982). Random-effects models for longitudinal data. *Biometrics 38*(4), 963–974.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 65*(1), 95–114.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(1), 3–36.

Wood, S. N. (2017). *Generalized Additive Models: an Introduction with R* (2 ed.). Boca Raton: Chapman & Hall/CRC.

Wood, S. N., N. Pya, and B. Säfken (2016). Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association 111*(516), 1548–1575.