# Online Appendix for Functional Generalized Additive Models

Article available in Journal of Computational and Graphical Statistics

Mathew W. McLean  Giles Hooker  Ana-Maria Staicu  Fabian Scheipl
David Ruppert

## A Identifiability Constraints

Notice that for $F^*(x,t) = F(x,t)+g(t)$, where $\int_{\mathcal{T}} g(t)\,dt = 0$ we have $\int_{\mathcal{T}} F^*(x,t)\,dt = \int_{\mathcal{T}} F^*(x,t)\,dt$ and thus we must impose constraints to ensure identifiability. If no constraints were used, the function $g(t)$ would be chosen to maximize the penalized log-likelihood given in Section 3.2 and $g(t)$ would be regularized by the difference penalties we use. The penalties alone are not enough to ensure identifiability, however. One possibility is to simply use a ridge penalty as in Marx and Eilers (1998). Due to the difference penalty used, functions of $t$ in the null space of the penalty are polynomials of degree $d_t - 1$. Therefore $d_t - 1$ constraints are necessary for identifiability. Instead of incorporating these constraints as we did for the constraint $\mathbf{1}^T \mathbb{Z}_{[,-1]}\boldsymbol{\theta} = 0$, which would require reparameterizing in terms of parametric and nonparametric components, we use the QR and singular value decomposition to numerically identify rank deficiency as outlined in Section 3.2.

For fixed smoothing parameters, different identifiability constraints yield the same predictions and the same estimated $\hat{F}(\cdot,\cdot)$ up to a constant, though different estimates for the variance of the estimated surface (and therefore different confidence bands) will be obtained. The GCV score is also invariant to the constraints used. It is possible to switch to an alternative set of constraints after fitting our model using a pivoted QR decomposition along the lines of Wood et al. (2012).

## B How to fit an FGAM in mgcv

Let $\mathbb{X}$ denote an $N \times J$ matrix of the observed measurements of the functional predictor, where $N$ is the number of sampled curves and $J$ is the number of measurements for each curve. Let $\mathbb{T}$ denote the $N \times J$ matrix of observation times for the predictor curves. Let $\mathbb{L}$ denote the $N \times J$ matrix of quadrature weights to use in our numerical integration of the surface $F(x,t)$ and let $\mathbf{y}$ be the $N$-vector of observed response values. The simplest FGAM (using all the package defaults) and without an intercept is specified in mgcv by

```
gam(y~te(X,T,by=L)-1),
```

where `te` specifies a tensor product smooth. The `-1` is included so that no intercept is fit and this is done here to simplify the notation. The variables in the `by` argument to `te` are treated as the covariates in a varying coefficient model. To make this association more explicit, a generalized varying coefficient model has the form (e.g., see Wood 2006b, p. 169)

$$g(\mu_i) = \theta_0 + f_1(x_{i1})x_{i2} + f_2(x_{i3}, x_{i4})x_{i5} + f_3(x_{i6})x_{i7} + \dots$$

As a special case, consider

$$g(\mu_i) = \theta_0 + f(x_{i1}, x_{i2})x_{i3} + f(x_{i1}, x_{i2})x_{i4} + \dots + f(x_{i1}, x_{i2})x_{i,J+2},$$

so each covariate, $x_{i3},\ x_{i4}, \dots, x_{i,J+2}$ has the same bivariate varying coefficient. Now suppose we have $x_{i1} \equiv X_i(t_j),\ x_{i2} \equiv t_{ij} \equiv t_j\ , x_{ip} \equiv l_{ij} \equiv l_j; p = j + 2; j = 1, \dots, J$ where the $l_j$'s are quadrature weights. Note that mgcv treats both variables $t$ and $l$ as if they depend on $i = 1, \dots, N$ though they do not for the FGAM. We now arrive at

$$g(\mu_i) = \theta_0 + \sum_{j=1}^{J} f(x_{i1}, x_{i2})x_{i,j+2} = \theta_0 + \sum_{j=1}^{J} f\{X_i(t_j), t_j\}l_j \approx \theta_0 + \int_{\mathcal{T}} f\{X_i(t), t\}\, dt,$$

so mgcv is fitting the model

$$E(Y_i|X_i) = \sum_{j=1}^{J} F(x_{ij}, t_{ij})l_{ij} = \sum_{j=1}^{J}\sum_{k=1}^{K_x}\sum_{m=1}^{K_t} \theta_{km} B_k^X(x_{ij}) B_m^T(t_{ij}) l_{ij},$$

where as in the paper, $B_k^X(\cdot)$ denotes the $k$th B-spline for the $x$-axis and $K_x$ is the dimension of the basis for $X$ (with equivalent definitions for the $t$-axis).

The matrix $\mathbb{B}_T$ which consists of $J \times K_x$ blocks of size $N \times K_t$ each is formed in mgcv. The $(i, j)$ entry in the $(m, n)$ block of $\mathbb{B}_T$ is given by $B_n^X(x_{im})B_j^T(t_{im})$. The design matrix used for the smooth is then the $NJ \times K_x K_t$ matrix

$$\mathbb{D} = \text{diag}[\text{vec}(\mathbb{L})]\mathbb{B}_T$$

The package enforces one constraint at this point because the row sums of the `by` variable matrix are constant ($\mathbb{L}\mathbf{1} = \mathbf{0}$). The constraint is $\mathbf{1}^T\mathbb{D}\boldsymbol{\theta} = \mathbf{0}$. How to implement this constraint during fitting and every other detail of the estimation procedure used by mgcv is outlined in Section 3.2 of the paper.

The default smoothing method for a tensor product smooth in mgcv is cubic regression splines, so the `bs` argument to `te` must be specified as `'ps'` for P-splines to be used. The `m` argument to mgcv specifies both the order of the spline and the order of the penalty. For P-splines, `m` can be specified as a list with length equal to the number of marginal bases. The argument `k` is a vector specifying the dimension of each marginal basis.

As an example, say we have an $N$-vector of responses $\mathbf{y}$, the $N \times J$ matrix of observed functional predictors $\mathbb{X}$ with observation times occurring at equally spaced points in $[0, 1]$, and that wish to use the midpoint rule aka rectangle method for approximating the integral. If we wish to use 10 cubic basis functions for the x-axis, 15 4th order basis functions for the t-axis, a second order difference penalty for the x-axis and a third order difference penalty for the t-axis, then the code to fit the FGAM with intercept is as follows

```
T=matrix( seq(0,1,l=J) ,N,J)
L=matrix(1/J,N,J)
fit=gam( y~te(X,T,by='L',bs='ps',k=c(10,15),m=list(c(2,2),c(4,3))) )
```

Note that in the documentation for P-spline smooths in mgcv (see `?p.spline`), it is noted that a smooth term of the form

```
s(x,bs="ps",m=c(2,3))
```

"specifies a 2nd order P-spline basis (cubic spline), with a third order difference penalty..." Though it is not standard for a cubic spline to be called 2nd order, this does seem to be what is implemented within `mgcv` and so we follow along with this specification.

Additional functional predictors are added by including additional `te` terms. Responses from other exponential family distributions are handled in the exact same way as the `glm` function in R.