# Generalized Additive Models

**Simon Wood**
School of Mathematics, University of Edinburgh, U.K.

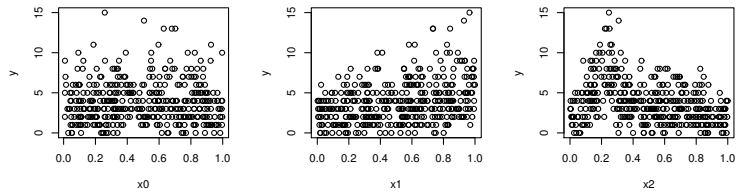
## The basic model

- The methods for simple smooth models can be used more generally, for example for *generalized additive models*.
- Response, $y_i$, predictors $x_{ji}$, model

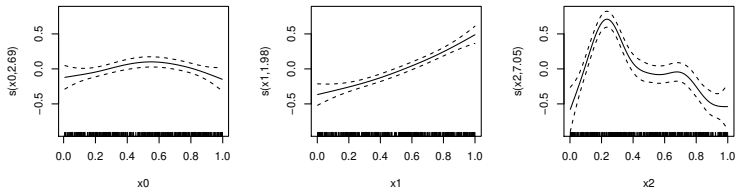
$$y_i \underset{\text{ind.}}{\sim} \pi(y_i|\mu_i, \boldsymbol{\theta}) \text{ where } g(\mu_i) = \mathbf{A}_i\boldsymbol{\gamma} + \sum_j f_j(x_{ji}).$$

  - $\pi$ is a p(d)f: location parameter $\mu$ and other parameters $\boldsymbol{\theta}$.
  - The $f_j$ are *smooth functions* to be estimated.
  - $\mathbf{A}$ is a model matrix: associated parameters $\boldsymbol{\gamma}$ to be estimated.
  - $g$ is a known *link function* (e.g. identity or log).
- If $\pi$ is an exponential family distribution then this is a GLM with linear predictor dependent on smooth functions of predictors.

## Example: Poisson regression

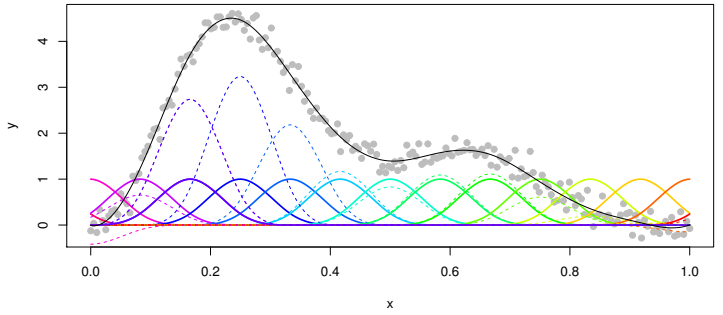



- $y_i \sim \text{Poi}(\mu_i)$ where $\log(\mu_i) = \alpha + f_0(x_{0i}) + f_1(x_{1i}) + f_2(x_{2i})$.
- `gam(y~s(x0)+s(x1)+s(x2),family=poisson())`




## Model representation and estimation

- Without $\sum f_j(x_{ji})$ the model is a standard regression model: use maximum likelihood estimation via Newton's method.
- With $\sum f_j(x_{ji})$ we:
  1. Represent each $f_j$ using its own basis expansion.
  2. Control $f_j$'s smoothness with its own smoothing prior/penalty.
  3. Estimate basis coefficients, smoothing parameters etc using penalized regression/empirical Bayes methods already covered.
- As previously basis expansion is $f_j(x) = \sum_k \beta_{jk} b_{jk}(x) \ldots$



## Model representation with basis

- The basis expansions for the $f_j$ turn the model into

$$y_i \underset{\text{ind.}}{\sim} \pi(y_i|\mu_i, \boldsymbol{\theta}) \text{ where } g(\mu_i) = \mathbf{X}_i\boldsymbol{\beta},$$

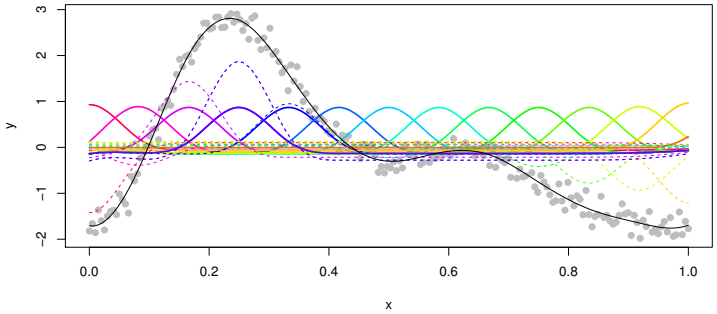
$\boldsymbol{\beta}^\mathsf{T} = (\boldsymbol{\gamma}^\mathsf{T}, \boldsymbol{\beta}_1^\mathsf{T}, \boldsymbol{\beta}_2^\mathsf{T} \ldots)$ and

$$\mathbf{X} = \begin{bmatrix} A_{11} & A_{12} & \cdots & b_{11}(x_{11}) & b_{12}(x_{11}) & \cdots & b_{21}(x_{21}) & \cdots \\ A_{21} & A_{22} & \cdots & b_{11}(x_{12}) & b_{12}(x_{12}) & \cdots & b_{21}(x_{22}) & \cdots \\ . & . & \cdots & . & . & \cdots & . & \cdots \\ . & . & \cdots & . & . & \cdots & . & \cdots \end{bmatrix}$$

- So the likelihood is readily computed and if $\pi$ is an exponential family distribution this is just a richly parameterized GLM.
- $\boldsymbol{\eta} = \mathbf{A}_i\boldsymbol{\gamma} + \sum_j f_j(x_{ji}) = \mathbf{X}\boldsymbol{\beta}$ is called the *linear predictor*.

## Identifiability

- The $f_j$ in $\sum_j f_j(x_{ji})$ are only identifiable up to an additive constant.
- So, impose identifiability constraints $\sum_i f_j(x_{ji}) = 0$, for all $j$.
- Can absorb into the bases. $b_{jk}(x) \leftarrow b_{jk}(x) - n^{-1}\sum_{i=1}^n b_{jk}(x_{ji})$, for all $k$ and drop least variable $b_{jk}(x)$ is one option.
- e.g. used on centred data...




- Note: no uncertainty about where a fully penalized straight line passes through zero.

## Smoothing penalty/prior

- With each $f_j$ we can associate a quadratic smoothing penalty $\lambda_j\boldsymbol{\beta}_j^\mathsf{T}\boldsymbol{\mathcal{S}}_j\boldsymbol{\beta}_j$ as in the univariate case.
- For notational convenience let $\mathbf{S}_j$ denote a matrix of zeroes with $\boldsymbol{\mathcal{S}}_j$ placed on one diagonal block so that $\boldsymbol{\beta}^\mathsf{T}\mathbf{S}_j\boldsymbol{\beta} = \boldsymbol{\beta}_j^\mathsf{T}\boldsymbol{\mathcal{S}}_j\boldsymbol{\beta}_j$.
- Writing $\mathbf{S}_\lambda = \sum_j \lambda_j\mathbf{S}_j$, the smoothing penalty for the GAM is now $\boldsymbol{\beta}^\mathsf{T}\mathbf{S}_\lambda\boldsymbol{\beta}$.
- Equivalently the smoothing prior is $\boldsymbol{\beta} \sim N(\mathbf{0}, \mathbf{S}_\lambda^-)$.
- The mathematical form of the penalty/prior is similar to the simple univariate case, so the same methods can be used.

## Inferential methods

- Penalized likelihood is optimized for the model coefficients $\boldsymbol{\beta}$, exactly as in the simple univariate case.
- Similarly, cross validation or Laplace approximate REML criteria are optimized to find the $\hat{\boldsymbol{\lambda}}$.
  - The only real complication is that we now have a multivariate optimization to perform over $\boldsymbol{\lambda}$.
- The effective degrees of freedom of a component $f_j$ is computed by summing those leading diagonal elements of the matrix $(\boldsymbol{\mathcal{I}} + \mathbf{S}_\lambda)^{-1}\boldsymbol{\mathcal{I}}$ corresponding to the coefficients $\boldsymbol{\beta}_j$.

## Computing the $\boldsymbol{\lambda}$ estimates

- Optimize the Laplace Approximate REML[1] (or other criterion) by Newton or Quasi- Newton method w.r.t. $\boldsymbol{\rho} = \log \boldsymbol{\lambda}$. i.e. maximize successive quadratic approximations to REML, based on derivatives of REML w.r.t. $\boldsymbol{\rho}$.
- Each trial $\boldsymbol{\rho}$ requires
  1. an inner Newton iteration to find $\hat{\boldsymbol{\beta}}$ for this $\boldsymbol{\rho}$, and hence evaluate the REML.
  2. an implicit differentiation step to find derivatives of $\hat{\boldsymbol{\beta}}$ w.r.t. $\boldsymbol{\rho}$ and hence the derivatives of the LAML.
- A less involved approach approximately maximizes the LAML by alternating updates of $\hat{\boldsymbol{\beta}}$ given $\boldsymbol{\lambda}$ with simple *Fellner-Schall*[2] updates of $\boldsymbol{\lambda}$ given $\hat{\boldsymbol{\beta}}$.

[1] Laplace Approximate Marginal Likelihood (or LAML), Wood, 2011, JRSSB
[2] Wood and Fasiolo, 2017, Biometrics

## Model selection tools

- We need means for comparing models/deciding what terms to include...
  1. Null space penalization: add an extra penalty and smoothing parameter for each $f_j$ which allows it to be penalized to zero during smoothing parameter estimation.
  2. P-values: 'invert' the Bayesian CI for $f_j$ to compute a p-value for $H_0 : f_j = 0$ (different for pure random effects terms).
  3. Akaike's Information Criterion becomes

$$-2l(\hat{\boldsymbol{\beta}}) + 2EDF$$

  but to use for model comparison, rather than $\boldsymbol{\lambda}$ estimation, we must correct for $\boldsymbol{\lambda}$ estimation uncertainty[3].
- In `mgcv`: 1. `gam(...,select=TRUE)` 2. `summary` or `anova` 3. `AIC`.

[3] Problem: Greven & Kneib 2010 Biometrika. Solution: Wood et al. 2016 JASA

## Model selection: null space penalties

- Smoothing parameter estimation does most of the work of model selection, by selecting between a large set of model functions of differing complexity.
- But most smoothing penalties have a null space of functions for which $\boldsymbol{\beta}_j^{\mathsf{T}} \boldsymbol{\mathcal{S}}_j \boldsymbol{\beta}_j = 0$. e.g. straight lines for the cubic spline.
- Hence no choice of $\lambda_j$ will penalize the term to zero.
- We can add an extra penalty (and smoothing parameter) to each term, made to only penalize functions in the penalty null space.
- How? Form eigen-decomp. $\boldsymbol{\mathcal{S}}_j = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathsf{T}}$, and let $\mathbf{U}_0$ denote the columns of $\mathbf{U}$ (eigenvectors) for which eigenvalues $\Lambda_{ii} = 0$.
- Then $\bar{\lambda}_j \bar{\boldsymbol{\mathcal{S}}}_j = \bar{\lambda}_j \mathbf{U}_0 \mathbf{U}_0^{\mathsf{T}}$ penalizes just the null space.
- If both $\lambda_j$ and $\bar{\lambda}_j \to \infty$ then $f_j \to 0$, penalized out of the model.

## Model selection: p-values

- Want to test $H_0 : f_j(x) = 0$.
- Given good frequentist coverage of Bayesian confidence intervals it is tempting to form Wald test statistic $\hat{\boldsymbol{\beta}}_j \mathbf{V}_j^{-1} \hat{\boldsymbol{\beta}}_j$, where $\mathbf{V}_j$ is Bayes covariance matrix for $\boldsymbol{\beta}_j$.
- Low power! Most heavily penalized components of $f_j$ are most heavily up-weighted by $\mathbf{V}_j^{-1}$.
- Use a low rank approximation to $\mathbf{V}_j$ in the Wald statistic, where rank is based on EDF of $\hat{f}_j$.
- Null distribution is then a sum of $\chi^2$ random variables: approximate p-value computable[4].
- Different approach needed for terms with no null space.

[4] not perfect - variability in other smoothing parameters neglected

## Simple model extensions

- Standard GLMs/GAMs cover single parameter exponential family distributions for $y$, notably Gaussian (normal), Poisson, binomial, gamma, and inverse Gaussian, plus quasi-likelihood models simply specifying $V$ such that $\text{var}(\mathbf{y}_i) = \phi V(\mu_i)$.
- The inference framework is not limited to these. `mgcv` also provides negative binomial, Tweedie, order categorical, censored normal...(`nb`, `tw`, `ocat`, `cnorm`, ...). See `?family.mgcv`.
- Occasionally the distribution of $y_i$ can change with $i$ - that is easily handled as well. See `?gfam`.
- Given that smooth functions can be viewed as Gaussian random effects, any random effect that makes a contribution $\mathbf{Zb}$ to a linear predictor, where $\mathbf{Z}$ is a model matrix for the term and $\mathbf{b} \sim \text{N}(\mathbf{0}, \mathbf{I}\lambda^{-1})$, can be treated just like any smooth function in the model (only p-value computation differs).
- Model terms like $z_i f_j(x_i)$, or terms with a separate smooth of $x$ for each level of a factor $g$ are also easy to include with no new methods needed. Both use the form `s(x,by=z)` in `mgcv`.

## Survival modelling

- Survival data can also be modelled in the same framework.
- Smooth Cox proportional hazards models are provided by the `cox.ph` family.
- Smooth Cox PH models for time varying covariates can also be handled using an equivalent Poisson likelihood trick, and some big data accelerations. See `?cox.pht`.
- Smooth accelerated failure time models are available via the censored normal family, `?cnorm`.

## Location scale and shape models[5]

- Actually, the methods are not restricted to only specifying a model relating $\mathbb{E}(y_i)$ to covariates. Other parameters of $y_i$'s distribution can also be modelled.
- Let $\boldsymbol{\theta}_i$ be the parameters of the distribution of $y_i$, often including the mean, $\mu_i$.
- We can model each element of $\boldsymbol{\theta}_i$ with its own linear predictor

$$y_i \underset{\text{ind.}}{\sim} \pi(y_i|\boldsymbol{\theta}_i) \text{ where } g(\theta_{ij}) = \mathbf{A}_{ij}\boldsymbol{\gamma}_j + \sum_k f_{kj}(x_{kji}).$$

- e.g. for a Gaussian, we might model the mean and the log standard deviation.
- In `mgcv` linear predictors are specified by supplying a list of formulae to `gam`. See `?family.mgcv` for distributions.

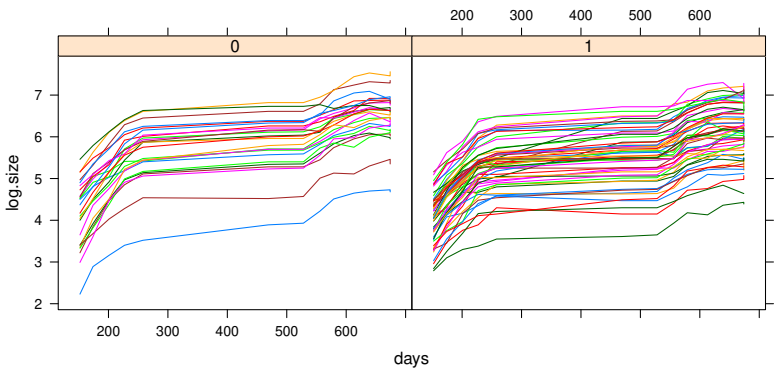
[5] Often known as GAMLSS (Rigby and Stasinopoulos, JRSSC, 2005), or *distributional regression*.

## GAMs with `mgcv:gam` in R

- Basically like any other regression model function in R.
- Modelling function `gam` has several key arguments:
  - a model formula: response on l.h.s and linear predictor on r.h.s.
    - the linear predictor can include smooth functions of predictors: e.g. `s(x,k=15,bs="cr")` is a rank 15 cubic spline.
    - if there are several linear predictors a list of formulae is supplied.
  - A `family`, specifying the distribution and any link functions.
  - A data frame containing the variables referred to in the formula.
- `gam` returns a fitted model object of class `gam`. Various methods functions are used to extract its components and summarize it...
  - `plot`, `gam.check`, `vis.gam`, `qq.gam`, `fitted`, `residuals` etc. are for visualization and checking.
  - `summary`, `anova`, `AIC`, `predict`, `vcov`, `gam.vcomp` etc. are for further inference and prediction.

# Example: Sitka spruce growth data

```
require(gamair); require(lattice); require(mgcv)
data(sitka); sitka$id.num <- factor(sitka$id.num)
xyplot(log.size~days|as.factor(ozone),data=sitka,
       type="l",groups=id.num)
```



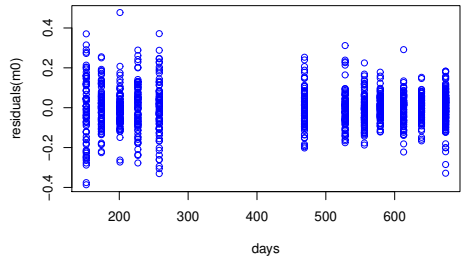

# Example: Sitka spruce growth model

- $\texttt{log.size}_i = f(\texttt{days}_i) + \gamma\texttt{ozone}_i + a_{\texttt{id}(i)} + b_{\texttt{id}(i)}\texttt{days}_i + \epsilon_i$
  $a_j \sim N(0, \sigma_a^2)$, $b_j \sim N(0, \sigma_b^2)$ and $\epsilon_i \sim N(0, \sigma^2)$.
- Fit with `mgcv` (family `gaussian` is default)

```
m0 <- gam(log.size ~ s(days) + s(id.num,bs="re")
      + s(id.num,days,bs="re")+ozone,data=sitka,method="REML")
```

- Basic checking with `gam.check(m0)` and `plot(m0)` and residual checks like...

```
plot(sitka$days,residuals(m0),xlab="days")
```




...variance not constant? Constant additive `ozone` effect?

# Example: Sitka spruce growth model 2

- $\texttt{log.size}_i = f(\texttt{days}_i) + \texttt{ozone}_i f_1(\texttt{days}_i) + a_{\texttt{id}(i)} + b_{\texttt{id}(i)}\texttt{days}_i + \epsilon_i$,
  $a_j \sim N(0, \sigma_a^2)$, $b_j \sim N(0, \sigma_b^2)$, $\epsilon_i \sim N(0, \sigma_i^2)$, $\log \sigma_i = f_2(\texttt{days}_i)$.
- In `mgcv`

```
m1 <- gam(list(log.size ~ s(days) + s(days,by=ozone) +
      s(id.num,bs="re") + s(id.num,days,bs="re"), ~ s(days)),
      family=gaulss,data=sitka, method="REML")
```

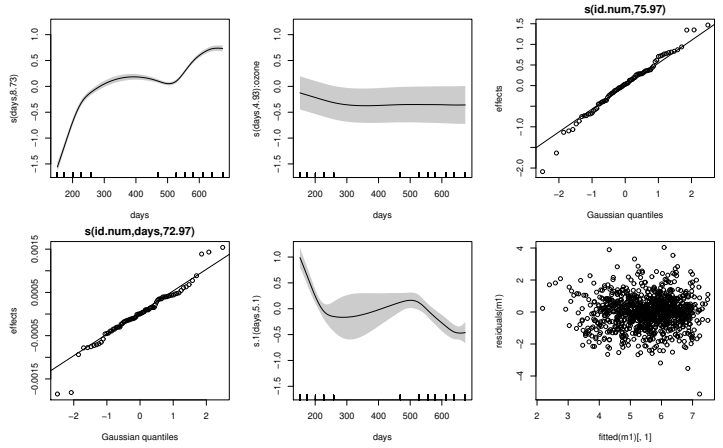
- AIC improves by about 180. Residual plots better.

```
> anova(m1)
...
Approximate significance of smooth terms:
                  edf Ref.df    Chi.sq  p-value
s(days)         8.733  8.955 2.239e+03  < 2e-16
s(days):ozone   4.933  5.752 2.751e+01 0.000106
s(id.num)      75.969 77.000 6.649e+06  < 2e-16
s(id.num,days) 72.971 77.000 1.675e+06  < 2e-16
s.1(days)       5.096  5.927 2.056e+02  < 2e-16
```

- Ozone effect significant (unlike if it's a constant). Also, dropping it increases AIC by 17.

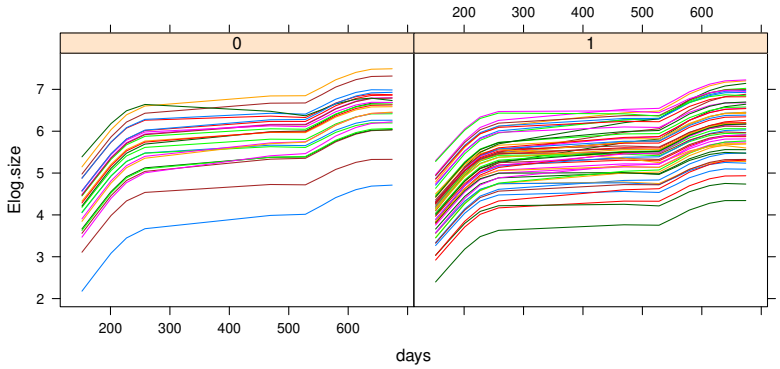
# Example: Sitka model 2 effects

```
par(mfrow=c(2,3),mar=c(4,4,2,2)); plot(m1,scheme=1)
plot(fitted(m1)[,1],residuals(m1))
```



## Example: Sitka model 2 predictions

```
sitka$Elog.size <- predict(m1)[,1]
xyplot(Elog.size~days|as.factor(ozone),data=sitka,type="l",
       groups=id.num)
```




## Summary

- GAMs allow a response to depend on smooth functions of predictor variables.
- The smooth functions are represented using a basis expansion and quadratic smoothing penalty.
- The quadratic penalties are equivalent to Gaussian priors on the coefficients, providing a Bayesian interpretation, including well behaved CIs.
- Basis coefficients are estimated by penalized MLE, smoothing parameters by REML or cross validation.
- A variety wide variety of response distributions is possible – for some we may provide linear predictors for other distribution parameters in addition to the mean.
- Model selection and checking are similar to any regression model (but check the basis dimensions).

## Model checking introduction

- As for any regression, examine standardised residuals to check for mean-variance and independence assumption violations.
- Details of the distribution beyond these properties are often less important (consider quasi-likelihood theory), but problems may have some influence on smoothness selection. See `qq.gam`.
- Careful residual plotting can indicate what is missing in a model.
- Are the smooth basis dimensions overly restrictive? Must check!
  - EDF close to its upper limit ($k'$, say) is *suspicious*.
  - Randomization test for residual pattern w.r.t. $x_j$: compare mean square difference between residuals for neighbouring $x_j$ values to mean square difference between randomly selected residual pairs. Pattern *may* indicate oversmoothing because basis too small.
  - `gam.check` provides such checks, amongst others. e.g. . .

```
        k' edf k-index p-value
s(x0) 9.0 2.5    1.04    0.77
```

- See `gam.check`, `residuals`, `fitted` etc. for more.