Some more advanced topics

Simon Wood
Mathematical Sciences, University of Bath, U.K.

Posterior simulation

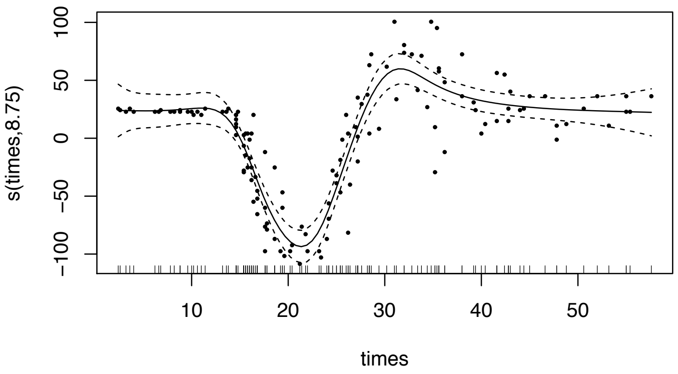
- Recall that for any fitted GAM we have the result

$$\boldsymbol{\beta}|\mathbf{y} \sim N(\hat{\boldsymbol{\beta}}, \mathbf{V}_\beta)$$

  (large sample approximation in the generalized case).
- This means that we can rapidly simulate from the posterior of any quantity derived from the fitted model.
- Such simulation is made much easier, if we can obtain the prediction matrix $\mathbf{X}_p$, mapping the model coefficents to the linear predictor, for any desired set of predictor variable values.
- `mgcv:predict.gam` computes such an $\mathbf{X}_p$ using `predict(...,type="lpmatrix")`

Posterior simulation example

- Here is an adaptive smooth fit to the motorcycle data.



- Suppose we would like a 95% CI for the trough to peak height.

Trough to peak CI

```
pd <- data.frame(times=seq(10,40,length=1000))
Xp <- predict(b,pd,type="lpmatrix") ## map coefs to fitted curves
beta <- coef(b);Vb <- vcov(b) ## posterior mean and cov of coefs
n <- 10000
br <- mvrnorm(n,beta,Vb) ## simulate n rep coef vectors from post.
a.range <- rep(NA,n)
for (i in 1:n) { ## loop to get trough to peak diff for each sim
  pred.a <- Xp%*%br[i,]  ## curve for this replicate
  a.range[i] <- max(pred.a)-min(pred.a) ## range for this curve
}
quantile(a.range,c(.025,.975))

    2.5%    97.5%
137.0796 174.5402
```

- This is very fast compared to boot-strapping, and less problematic.
- The for loop is only for clarity, it can be eliminated.

Correlated data

- Correlated data can be modelled using high rank Gaussian random fields (smoothers), or by GEE type assumption of a covariance structure for the response.
- For Gaussian data it is straightforward to incorporate a known correlation structure into the likelihood. If such a structure is sparse (it's Choleski factor, or inverse Choleski factor is sparse) then efficient computation is sometimes possible.
- An AR1 model is an example of such a sparse structure.
- Unknown correlation parameters can be optimized numerically, or by simple profile likelihood grid search.
- Software for correlated data is a bit limited at the moment (but if you don't have too many smooth terms, check out INLA).

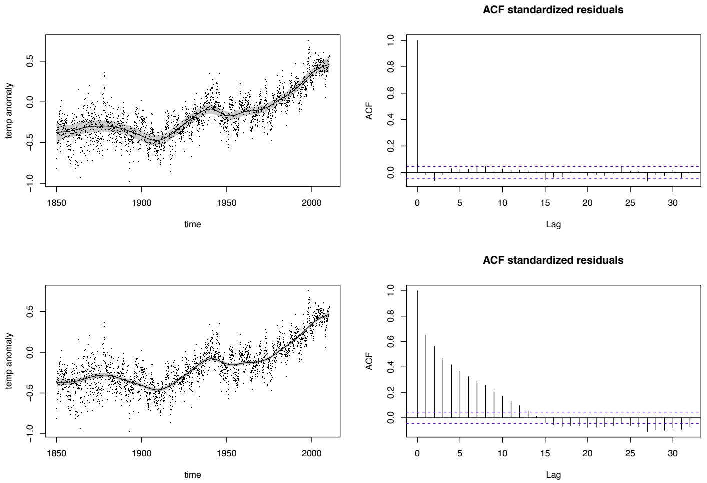
GAM + AR1 example

- The Hadley Centre (UK) assembles monthly global mean temperature datasets, going back to 1850 (e.g. hadcrut3 from their web site).
- The data appear quite noisy, so it is important to be able to say, objectively, what the underlying smooth trend in the data looks like.
- There is an annual cycle in the data, essentially because the Northern and Southern Hemispheres respond differently to incoming solar radiation.
- A reasonable model, of temperature anomaly, a_i, is

$$a_i = f(t_i) + g(m_i) + e_i$$

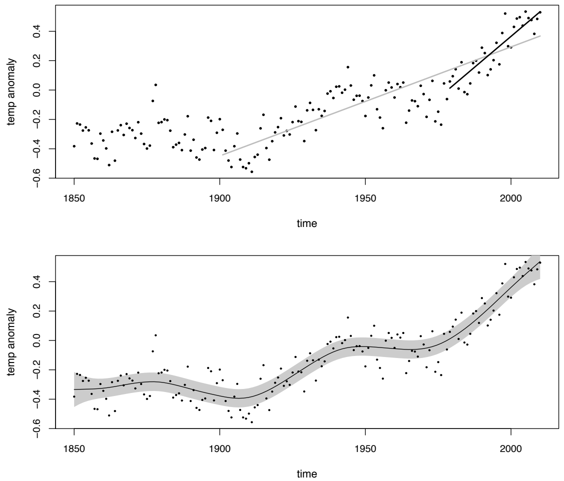
where the e_i are AR1 gaussian errors, with unknown correlation parameter. $g(m_i)$ is cyclic function of month.

`bam(...,rho=.98)` fit of AR1 GAM



- Upper is fit with REML optimal correlation parameter, lower is equivalent model without auto-correlation, but forced to have same smoothing parameters.

Annual data



- Top is what is currently done for IPCC presentation to policy makers.

Spatial correlation

- Sometimes a relatively high rank smooth suffices (e.g. a thin plate spline of space).
- Sometimes `bam` can be more efficient than `gam` for such high rank terms, but much over rank 1000 and the methods become impractically slow.
- A GEE type approach to correlation can be used with `gamm` via `nlme` type correlation structures, but convergence is not very reliable.
- Essentially the approach assumes a parameterized correlation structure for the working data used at each PQL iteration during fitting (of course this is just a likelihood method if the response is Gaussian).

Spatial correlation example

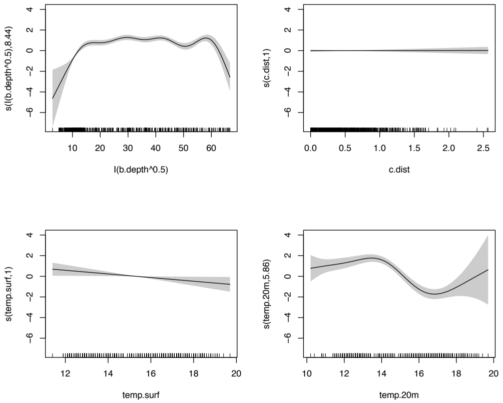
- Revisiting the fish egg data, from earlier, we could try to force more of the explanatory power onto the covariates by replacing the spatial smooth with an assumption about residual spatial autocorrelation. Let's assume a simple model in which correlation decays as a half Gaussian. . .

```
mack$lon <- mack$lon + (runif(n)-.5)/20 ## jitter location

gmm <- gamm(egg.count ˜ s(I(b.depthˆ.5)) + s(c.dist) +
            s(temp.surf) + s(temp.20m)+offset(log.net.area),
            data=mack,family=quasipoisson,
            correlation=corGaus(.1,form=˜lon+lat))
```

- See `nlme` documentation for more on the correlation structure.
- Fitting takes 10s of minutes. . .

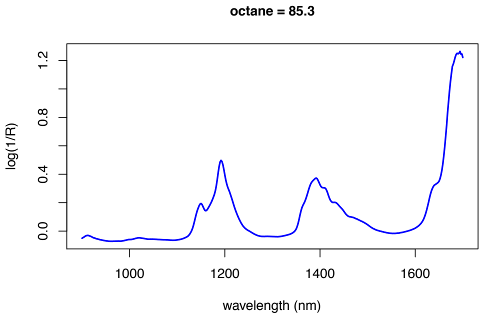
CorGaus GAM effects



- The sea bed depth effect is much stronger in this model.
- Spatial correlation in these models is an active area of research.

Functional data

- Function on scalar, and scalar on function regressions can readily be cast as GAMs/ penalized GLMs.
- Start with scalar on function and consider predicting octane rating from near infrared spectrum of gasoline.



scalar on function

- There are 60 such spectrum ($k_i(x)$) - octane (y_i) pairs (x is wavelength), and a model might be

$$y_i = \alpha + \int f(x)k_i(x)dx + \epsilon_i \simeq \alpha + \frac{1}{h}\sum_{k=1}^{p} k_i(x_k)f(x_k) + \epsilon_i$$

  where $f(x)$ is a smooth function of wavelength, and the x_k are evenly spaced h apart.

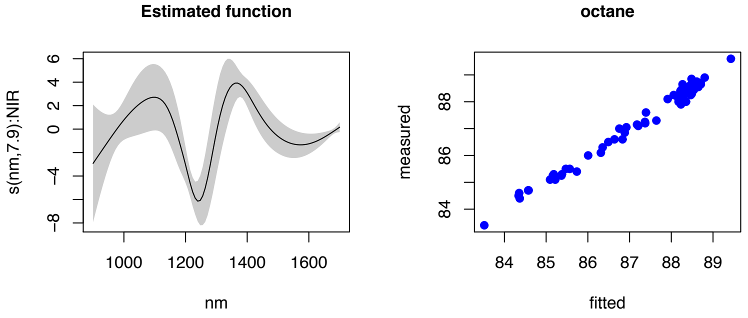
- Let $X_{ik} = x_k \; \forall \; i$ and $L_{ik} = k_i(x_k)/h$. In `mgcv:gam`

```
s(X,by=L)
```

  evaluates $\sum_k f(X_{ik})L_{ik} = \frac{1}{h}\sum_{k=1}^{p} k_i(x_k)f(x_k)$, by invoking a summation convention for matrix arguments of smooths (including `te/2`).

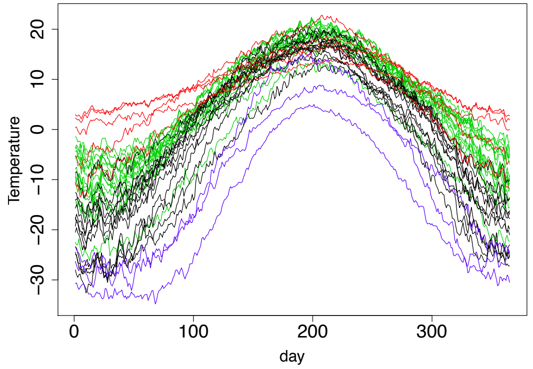
Octane fit

```
library(pls);data(gasoline);gas <- gasoline
nm <- seq(900,1700,by=2) ## create wavelength matrix...
gas$nm <- t(matrix(nm,length(nm),length(gas$octane)))
b <- gam(octane˜s(nm,by=NIR,bs="ad"),data=gas)
plot(b,rug=FALSE,shade=TRUE,main="Estimated function")
plot(fitted(b),gas$octane,...)
```



function-on-scalar

- Annual temperature data from some Canadian locations.

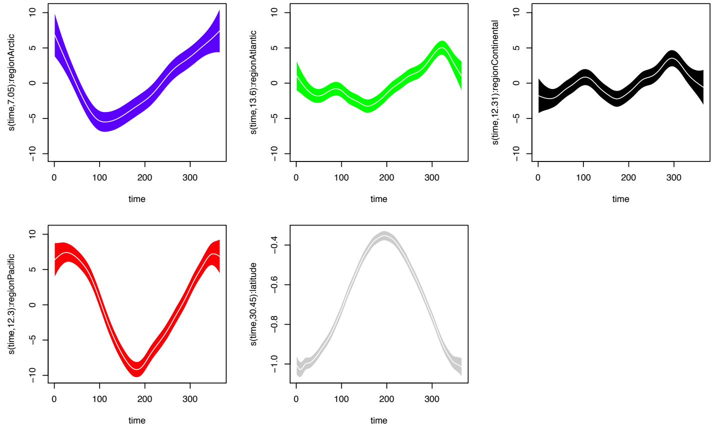


- Colour denotes region: blue is Arctic, black continental, red Pacific, green Atlantic.
- Model: if profile from region j:

$$\texttt{temp}_i = f_j(\texttt{t}_i) + f(\texttt{t}_i)\texttt{latitude}_i + \epsilon(\texttt{t}_i)$$

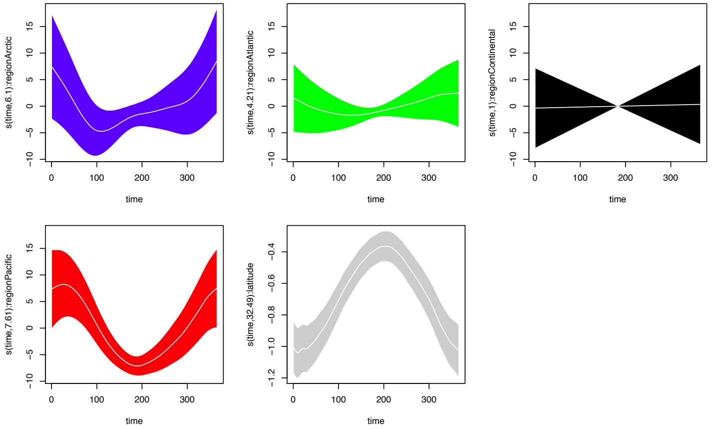
i.i.d error fit

```
b <- gam(T˜region+s(time,k=20,bs="cr",by=region)+
           s(time,k=40,bs="cr",by=latitude),
           data=dat,method="REML")
```



AR1 error fit

```
b1 <- gamm(T~region+s(time,k=20,bs="cr",by=region)+
             s(time,k=40,bs="cr",by=latitude),
             data=dat,correlation=corAR1(form=~1|place))
```



The end.