Checking & Selecting GAMs

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

Model checking overview

- Since a GAM is just a penalized GLM, residual plots should be checked exactly as for a GLM.
- It should be checked that smoothing basis dimension is not restrictively low. Defaults are essentially arbitrary.
- The GAM analogue of co-linearity is often termed 'concurvity'. It occurs when one predictor variable could be reasonably well modelled as a smooth function of another predictor variable. Like co-linearity it is statistically destabilising and complicates interpretation, so is worth checking for.

Residual checking

- Deviance, Pearson, working and raw residuals are defined for a GAM in the same way as for any GLM.
- In `mgcv` the `residuals` function will extract them, defaulting to deviance residuals.
- Residuals should be plotted against
  1. fitted values.
  2. predictor variables (those included and those dropped).
  3. time, if the data are temporal.
- Residual plotting aims to show that there is something wrong with the model assumptions. It's good to fail.
- The key assumptions are
  1. The assumed mean variance relationship is correct, so that scaled residuals have constant variance.
  2. The response data are independent, so that the residuals appear approximately so.

Distribution checking

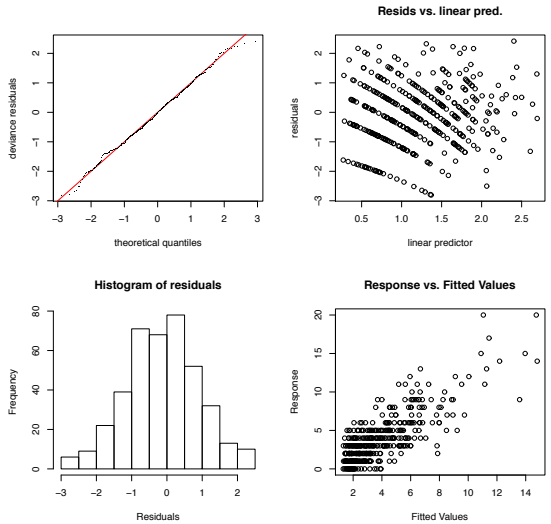
- If the independence and mean-variance assumptions are met then it is worth checking the distributional assumption more fully.
- The implication of quasi-likelihood theory is that provided the mean variance relationship is right, the other details of the distribution are not important for many inferential tasks.
- QQ-plots of residuals against standard normal quantiles can be misleading in some circumstances: for example low mean Poisson data, with many zeroes.
- It is better to obtain the reference quantiles for the deviance residuals by repeated simulation of response data, and hence residuals, from the fitted model. `mgcv` function `qq.gam` will do this for you.
- `gam.check` produces some default residual plots for you.

Residual checking example

```
> b <- gam(y~s(x0)+s(x1,x2,k=40)+s(x3)+s(x4),
+ family=poisson,data=dat,method="REML")
>
> gam.check(b)

Method: REML   Optimizer: outer newton
full convergence after 8 iterations.
Gradient range [-0.0001167555,3.321004e-05]
(score 849.8484 & scale 1).
Hessian positive definite, eigenvalue range [9.66288e-05,10.52249].

[edited]
```

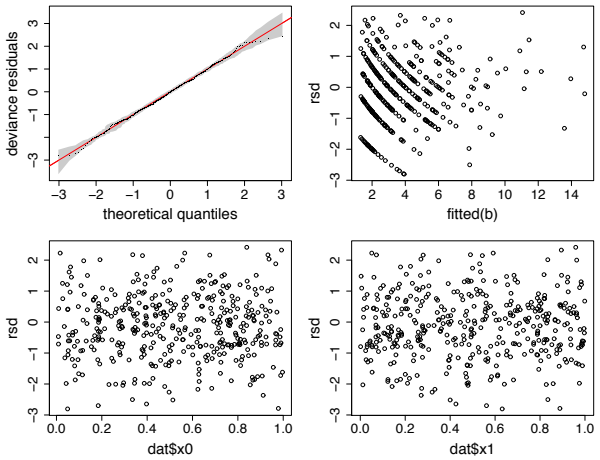
- The printed output is rather detailed information about smoothing parameter estimation convergence.
- 4 residual plots are produced, the first is from `qq.gam`, unless quasi-likelihood is used, in which case we have to fall back on a normal QQ-plot (but anyway don't care about this plot). The rest are self explanatory.

`gam.check` plots



More residual plots

```
rsd <- residuals(b)
qq.gam(b,rep=100); plot(fitted(b),rsd)
plot(dat$x0,rsd); plot(dat$x1,rsd)
```



Checking *k* the basis dimension

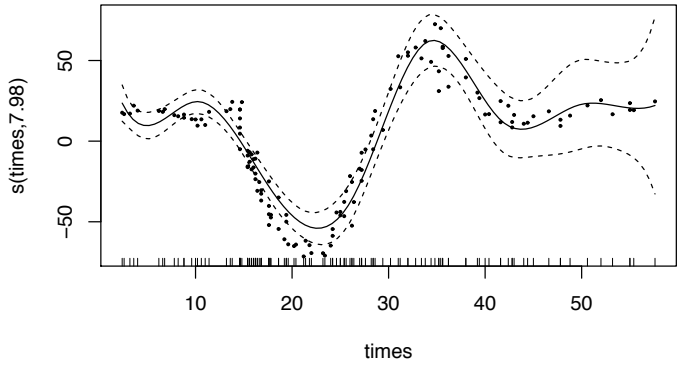
- Provided it is not restrictively low the choice of basis dimension, *k*, is not critical, because the effective degrees of freedom of a term are primarily controlled by the smoothing penalty.
- But it *must* be checked that *k* is *not* restrictively low — default values are arbitrary.
- Four checking methods are useful.
  1. Plot *partial residuals* over term estimates, looking for systematic departures.
  2. Test the residuals for residual pattern.
  3. Try re-smoothing the model deviance residuals with respect to the covariate(s) of interest using a higher *k*, to see if any pattern is found.
  4. Try re-fitting the model with increased *k* and see if the smoothness selection criterion increases substantially.
- 1 and 2 should be routine. 3 and 4 are useful if you are suspicious, but are also more time consuming.

Partial residuals

- Partial residuals are specific to each smooth term.
- Recall that the *working residuals* for a GLM are the weighted residuals from the working linear model using in the IRLS fitting scheme, at convergence.
- The partial residuals for f_j are the working residuals that you obtain using a linear predictor with $\hat{f}_j$ set to zero. These are the same as the working residual added to $\hat{f}_j$.
- The partial residuals should look like a random scatter around the smooth.
- Systematic deviation of the mean partial residual from $\hat{f}_j$ can indicate that k is too low.

Partial residual example

```
library(MASS)
m <- gam(accel~s(times,bs="ps"),data=mcycle,weights=w)
plot(m,residuals=TRUE,pch=19,cex=.3)
```



. . . note the systematic pattern in the departure of the partial residuals from the smooth. Should increase k.

A simple residual test

- An estimate of scale parameter ϕ can be obtained by differencing scaled residuals.
- Differencing residuals that are neighbours according to some covariate(s) should give an estimate of ϕ that is statistically indistinguishable from a differencing estimate obtained with any random ordering of residuals, *if there is no residual pattern with respect to the covariates*.
- This is the basis for a simple, and rapid, randomisation test.
- If pattern is detected, then it may indicate that k is too low.
- . . . but care is needed: pattern may also be caused by mean-variance problems, missing covariates, structural infelicities, zero inflation etc. . .

Residual test example

```
> gam.check(m)
[edited]
Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.

            k'   edf k-index p-value
s(times) 9.000 7.981   0.529       0
```

- k-index is ratio of neighbour differencing scale estimate to fitted model scale estimate.
- k' is the maximum possible EDF for the term.
- Here a low p-value coupled with high EDF suggests k may be too low.

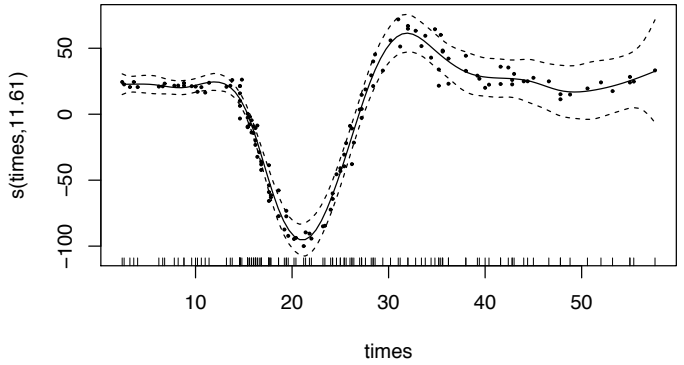
Alternative check example

```
> rsd <- residuals(m)
> ## smooth residuals with k doubled, to check pattern
> ## gamma>1 favours smoother models.
> gam(rsd~s(times,bs="ps",k=20),data=mcycle,gamma=1.4)

Family: gaussian
Link function: identity

Formula:
rsd ~ s(times, bs = "ps", k = 20)

Estimated degrees of freedom:
12.875  total = 13.87489

GCV score: 83.21058
```

This approach is not really needed for a single term model, but is usefully efficient, relative to refitting with larger k, when there are many terms present.

k fixed

```
m <- gam(accel~s(times,bs="ps",k=20),data=mcycle,weights=w)
```



- Further check now find no suggestion that k is too low.
- There are some differences in the k required with different bases. The default ”tp” basis gives acceptable results with k=10 (it is designed to be the optimal basis at a given k).

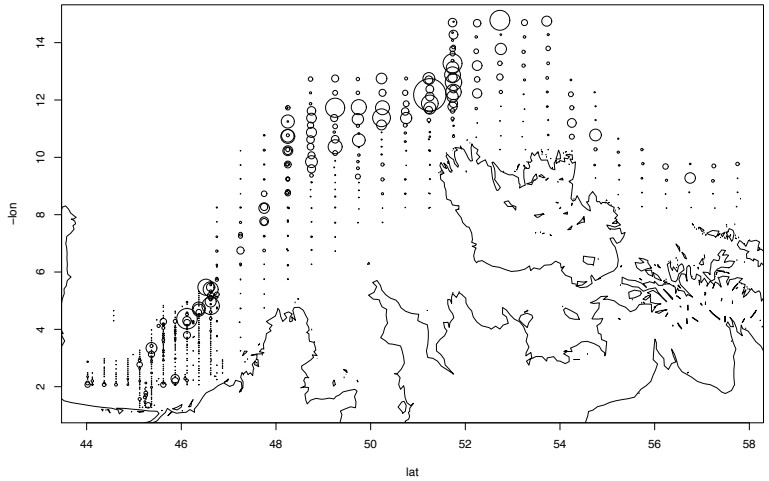
Concurvity

- Consider a model containing smooths f_1 and f_2.
- We can decompose $f_2 = f_{12} + f_{22}$ where f_{12} is the part of f_2 representable in the space of f_1, while f_{22} is the remaining component, which lies exclusively in the space of f_2.
- A measure of concurvity is $\alpha = \|f_{12}\|^2/\|f_2\|^2$, leading to 3 estimates
  1. $\hat{\alpha} = \|\hat{f}_{12}\|^2/\|\hat{f}_2\|^2$.
  2. The maximum value that α could take for any estimates, using the given bases for f_1 and f_2.
  3. The ratio of the ‘size’ of the basis for f_{12} relative to the basis for f_2, using some matrix norm.
- Function `concurvity` reports 1 as ’observed’, 2 as ’worst’ and 3 as ’estimated’. All are in $[0, 1]$.
- The measure generalizes to more components.

Concurvity consequences

- Concurvity can make interpretation difficult.
- Spatial confounding is a common example: you need a spatial effect in the model, but all the other covariates are somehow functions of space.
- A technical problem is that smoothing parameter estimates may become highly correlated and variable, which degrades the performance of inferential methods that are conditional on those estimates (confidence intervals and p-values).
- Under ML or REML smoothness selection `sp.vcov` and `gam.vcomp` can help diagnose this problem.
- Model averaging over the sampling distribution of the smoothing parameters can help in severe cases.

Concurvity/Spatial confounding example



`concurvity` example

```
library(gamair)
data(mack)
gm <- gam(egg.count~s(lon,lat,k=100)+s(I(b.depth^.5))+s(salinity)+s(temp.20m)
          +offset(log.net.area),data=mack,family=quasipoisson,method="REML")
concurvity(gm)
                  para s(lon,lat) s(I(b.depth^0.5)) s(salinity) s(temp.20m)
worst    1.063513e-17  0.9899778         0.9874163   0.9300386   0.9621984
observed 1.063513e-17  0.8308139         0.9518048   0.9232639   0.8736039
estimate 1.063513e-17  0.5500618         0.9360886   0.8952500   0.9294740
```

- This output shows the concurvity of each term with all the other terms in the model. Basically space is confounded with everything.
- With spatial confounding it sometimes helps to increase the smoothing parameter for space, e.g. until the REML score is just significantly different to its maximum.

```
gm1 <- gam(egg.count~s(lon,lat,k=100,sp=0.05)+s(I(b.depth^.5)) +s(salinity)+s(temp.20m)
           +offset(log.net.area),data=mack,family=quasipoisson,method="REML")
```

Model selection

- A large part of what would usually be thought of as model selection is performed by smoothing parameter estimation, but smoothing selection does not usually remove terms altogether.
- There are three common approaches to deciding what terms to include.
  1. Get smoothing parameter estimation to do all the work, by adding a penalty for the un-penalized space of each term.
  2. Compute approximate p-values for testing terms for equality to zero, and use conventional selection strategies (backwards, forwards, backwards-forwards, etc).
  3. Use similar strategies based on AIC, or on the GCV or ML scores for the model.

Penalizing the penalty null space

- The penalty for a term is of the form $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}\boldsymbol{\beta}$.
- Usually $\mathbf{S}$ is not full rank so some finite (M) dimensional space of functions is un-penalized.
- In consequence penalization can not completely remove the term from the model.
- Consider eigen-decomposition $\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}$. The last M eigenvalues will be zero. Let $\tilde{\mathbf{U}}$ denote their corresponding eigenvectors.
- $\boldsymbol{\beta}^{\mathrm{T}}\tilde{\mathbf{U}}\tilde{\mathbf{U}}^{\mathrm{T}}\boldsymbol{\beta}$ can be used as an extra penalty on just the component of the term that is unpenalized by $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}\boldsymbol{\beta}$.
- Adding such a penalty to all the smooth terms in the model allows smoothing parameter selection to remove terms from the model altogether.

Null space penalization in action

```
> gm <- gam(egg.count~s(lon,lat,k=100)+s(I(b.depth^.5))+
+           s(c.dist) + s(temp.surf)
+           +s(salinity)+s(temp.20m)+offset(log.net.area),
+           data=mack,family=quasipoisson,method="REML",select=TRUE)
> gm

Family: quasipoisson
Link function: log

Formula:
egg.count ~ s(lon, lat, k = 100) + s(I(b.depth^0.5)) + s(c.dist) +
    s(temp.surf) + s(salinity) + s(temp.20m) + offset(log.net.area)

Estimated degrees of freedom:
60.60  2.17  0.42  0.00  1.83  5.17  total = 71.19

REML score: 515.0758
```

- So `temp.surf` is penalized out, and `c.dist` nearly so!

summary(gm)

```
Family: quasipoisson
Link function: log

...

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.9506     0.1237   23.85   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
                   edf Ref.df     F  p-value
s(lon,lat)        61.280 73.602 3.094 5.28e-13 ***
s(I(b.depth^0.5))  2.593  3.164 3.154  0.02354 *
s(c.dist)          1.000  1.000 1.532  0.21688
s(temp.surf)       1.000  1.000 0.133  0.71597
s(salinity)        1.001  1.001 8.891  0.00313 **
s(temp.20m)        5.960  6.941 3.504  0.00136 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.825   Deviance explained = 90.2%
REML score = 510.79  Scale est. = 4.4062    n = 330
```

p-values and all that

- A p-values for smooth term, f, with a finite dimensional un-penalized space can be computed by a rather involved inversion of the Bayesian intervals for a smooth, which give good frequentist performance.
- The test statistic is $\hat{\mathbf{f}}^{\mathrm{T}}\mathbf{V}_f^{\tau-}\hat{\mathbf{f}}$ where $\mathbf{V}_f^{\tau'-}$ is a generalized rank τ' pseudoinverse of the Bayesian covariance matrix for $\mathbf{f}$ the vector of f evaluated at the observed covariate values. τ' is a version of the effective degrees of freedom of $\hat{f}$, based on $2\mathbf{F} - \mathbf{FF}$ in place of $\mathbf{F}$.
- For random effects, and smooths with no un-penalized space, another approach is needed.
- In both cases the p-values are conditional on the smoothing parameter estimates (you have been warned!)
- Refitting the egg model without null space penalization and calling `summary(gm)` gives...

generalized AIC etc

- An approximate AIC is often used for model selection:

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

  where $\hat{\beta}$ are the maximum *penalized* likelihood estimates and τ is the effective degrees of freedom of the whole model, and the UBRE (Mallows C_p) score used for smoothness selection for known scale parameter is directly proportional to this.
- AIC usually gives very similar results to selecting models on the basis of the GCV (or UBRE) score.
- The ML score can also be used in the same way, but not REML (because of the usual lack of comparability between models with different fixed effect structures).

GLRT via `anova`

- Approximate generalized likelihood ratio testing can also be performed, again based on the maximum penalized likelihood estimates and effective degrees of freedom, and again, conditional on the smoothing parameter estimates.
- The `anova` function in R can be used for this purpose.
- The approximation has limited justification. If the model terms can all be closely approximated by unpenalized terms, then the approximation is often reasonable, but note that random effects can not be approximated in this way, and the approximation breaks down in this case.
- Unless your smooths are really frequentist random effects, resampled from their prior/marginal with every replication of the data, then a GLRT (or AIC) based on the ML or REML score is a bad idea.

Summary

- Model checking is just like for a GLM + *check that smoothing basis dimensions are not too small*.
- Concurvity is the generalization of co-linearity to worry about in interpretation.
- A variety of model selection tools are available, including full penalization, generalized AIC, term specific p-values and approximate GLRT tests.
- Tests/p-values are approximate and conditional on smoothing parameter estimates.
  1. When smoothing parameter estimators are highly correlated (see e.g.`sp.vcov`), single term p-values should be treated with caution.
  2. GLRT tests are a particularly crude approximation, and can fail completely when random effects are involved.
- GAMs are statistical models and there are reasonable statistical tools available to help in the process of model building, but if you want machine learning, GAMs are probably not the place to start.

Additive versus Interaction

- $f_1(x) + f_2(z)$, $f_3(x, z)$ or $f_1(x) + f_2(z) + f_3(x, z)$?
- Conceptually $f_1(x) + f_2(z)$ appears nested in $f_3(x, z)$, but unless you choose the smoothing penalties *very* carefully it won't be.
- The `t2` tensor product construction in `mgcv` build smooths with penalties that do nest $f_1(x) + f_2(z)$ in $f_3(x, z)$ (basically following a reduced rank version of the SS-ANOVA approach of Gu and Wahba), but the price you pay is the need to use penalties that are somewhat un-intuitive. `pen.edf` and `gam.vcomp` are useful with such terms.
- A simpler approach simply removes the lower order interactions from a higher order basis. `ti` terms in `mgcv` do this.