

Review: Hierarchical generalized additive models: an introduction with mgcv

Fabian Scheipl

November 12, 2018

Basic reporting

p. 23: The discussion on model comparison is very reasonable and pragmatic as it is. A reference to model averaging for maximizing predictive accuracy would also be useful and appropriate here (e.g. DOI:10.1002/ecm.1309), I think, as would be mentioning that "ts" and similar no-penalty-nullspace bases or using `select = TRUE` can also be used for variable/model selection in `mgcv` (see DOI:10.1016/j.csda.2011.02.004).

p. 41: A fairly recent review of functional regression methods is Morris, Jeffrey S.

"Functional regression."

Annual Review of Statistics and Its Application 2 (2015): 321-359.

DOI:10.1146/annurev-statistics-010814-020413,

I think this would be the more relevant reference than Scheipl/Staicu/Greven or Ramsay/Silverman here. The discussion paper at DOI:10.1177/1471082X16681317 gives a more up-to-date over `mgcv` and boosting based functional regression approaches of the refund/Greven group.

Experimental design

Section V:

It seems very likely that, for any given true data generating process, the probability of over- or undersmoothing depends crucially on the noise level of the data: a priori, I would have expected noisier data to tend to yield smoother functions (in Bayesian terms: a less concentrated likelihood will yield a posterior closer to the prior than a more concentrated one). This seems to agree with the reported results for these almost noise-free data but the authors should verify that their conclusions hold across multiple replicates at different noise levels, not just the single one that was used here. The DGP they used gives a correlation > 0.95 between the true additive predictor (signal) and the observed responses, or, in other words a signal-to-noise ratio above 10:

```
with(dat, var(y)) / 0.2^2 # SNR
# [1] 11.61404
with(dat, cor(y, sin(freq*x))) # Cor(Response, Signal)
# [1] 0.9537038
```

If I increase the noise to `sigma = .5` or `1`, the pattern of over-/underfitting changes in the predictable way (much less overfitting of low-frequency terms, more oversmoothing of high-frequency terms for the mis-specified model), see figures below.

If the authors want to draw reliable conclusions on `mgcv`'s tendency to over/or undersmooth group-specific smooths for certain model (mis-)specifications, they should

1. repeat the simulation with intermediate and low signal-to-noise ratios to check that their results still hold then (... they won't, I think)
2. do this on replicated data, as no reliable conclusions can be drawn from a single replicate.

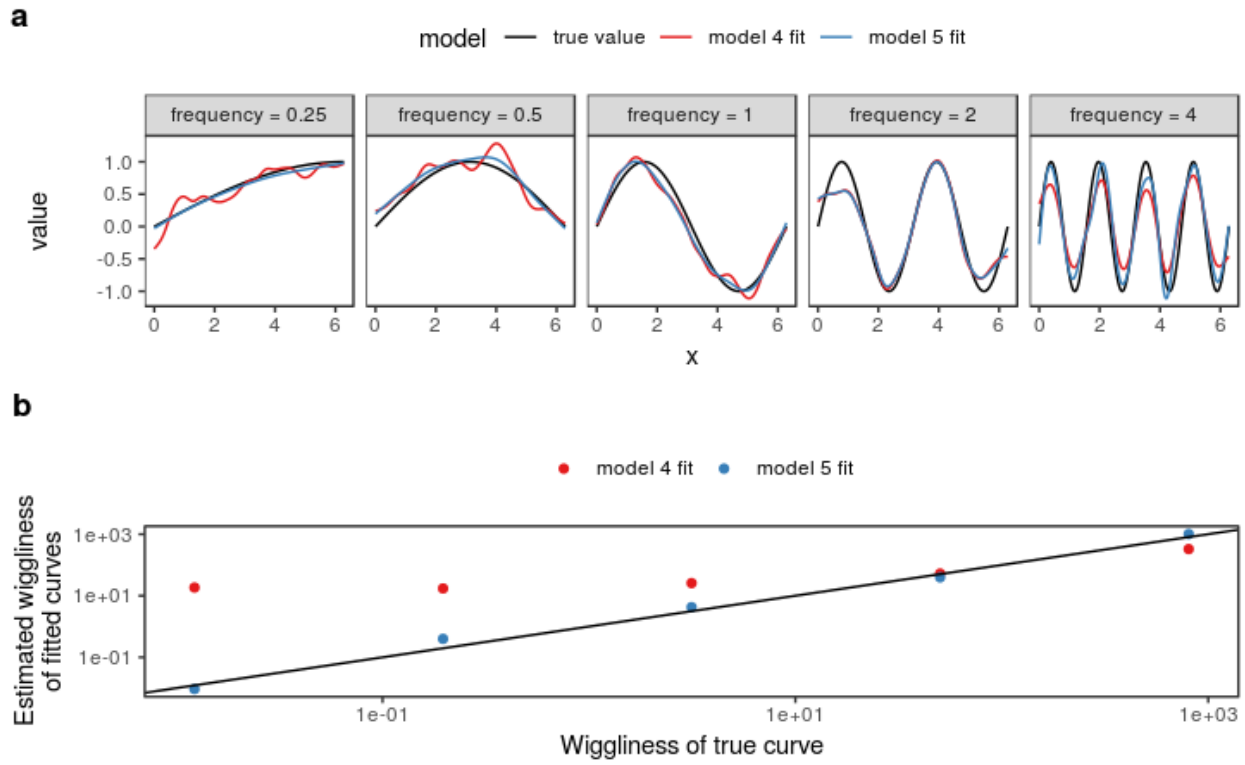


Figure 1: $\sigma = .5$

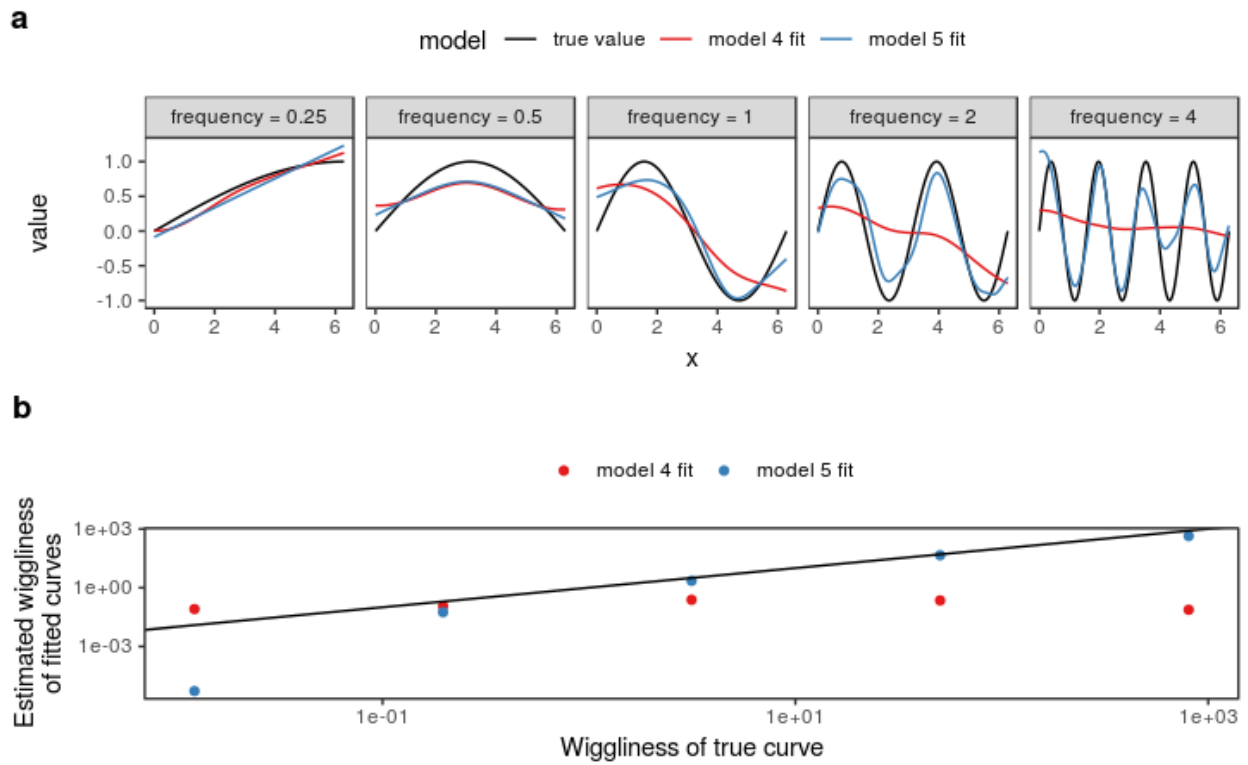


Figure 2: $\sigma = 1$

“This implies that assuming equal smoothness will result in underestimating the true smoothness of low-variability terms, and thus lead to more variable estimates of these terms.” is not a conclusion supported by the evidence provided by the authors, it is too broad and too confident.

Validity of the findings

Table 1: If I rerun this, I get a slightly lower AIC for model 2...?

```
AIC_table
## A tibble: 10 x 4
#   Model      df  AIC deltaAIC
# [...]
# 6 bird_mod1    51 3374    1819
# 7 bird_mod2   141 1556         2
# 8 bird_mod3   208 1680    125
# 9 bird_mod4   130 1555         0
# 10 bird_mod5  197 1635    81
## (see session info below.)
```

(Discussion of) Table 1 : “For the bird_move dataset (Table [1]B), model 2 (global smoother plus group-level smoothers with a shared penalty) fits the data best (which is good as we simulated the data from a model with this structure!)”

Not in terms of AIC – it may have the higher likelihood:

```
logLik(bird_mod2)
# 'log Lik.' -637.2433 (df=140.961)
logLik(bird_mod4)
# 'log Lik.' -647.1986 (df=130.1435)
```

but AIC is lowest for model 4, so this remark should be qualified with the additional likelihood information. In general I would conjecture that, if “enough” information is available to reliably estimate trends for every group level, model 4 will have a better AIC than model 2 as it’s the more parsimonious model specification of two models with essentially the same capacity (as you explain yourself later, the global trend is completely concave with the group-specific smooths) and will thus achieve similar goodness-of-fit at lower model complexity...?

p. 25/ Fig. 14: `mgcv::qq.gam` actually provides very good facilities for proper checks of the distributional assumptions that don’t just eyeball whether/where the dots are “too far” from the diagonal. I would prefer to see QQ-plots with simulation-based reference lines here. This is (much) more important for discrete responses, but should be described as best practice, nonetheless. Even better would be to mention / use the model diagnostics implemented in Fasiolo’s new `mgcViz` package, if its functionality extends to the kinds of models used here – I think it does, though.

p. 27: Table 2, 3: RMSE seems like a rather poor choice of evaluation metric to me here, as it will be dominated by the errors made on the largest observations or fitted values. Why not use the criterion used to fit the model, i.e. compute (average) predictive likelihoods / deviances, instead? That would more accurately reflect predictive performance over all orders of magnitude of predictions and observations, I think. The RMSE may be more interpretable/intuitive, but it doesn’t give reasonable answers for response distributions that are this skewed and where the variance increases quadratically (!) with the mean.

Fig. 15, 16: not using `scales = "free_y"` and adding `+ scale_y_log10()` gives a more informative and legible plot for the different species / lakes, I think. As would including the test set data in a different shape/colour (or even: showing just the test data instead of the training data).

p. 29: “We will exclude the `gam.check` diagnostic plots and results, as they do not indicate any issues with

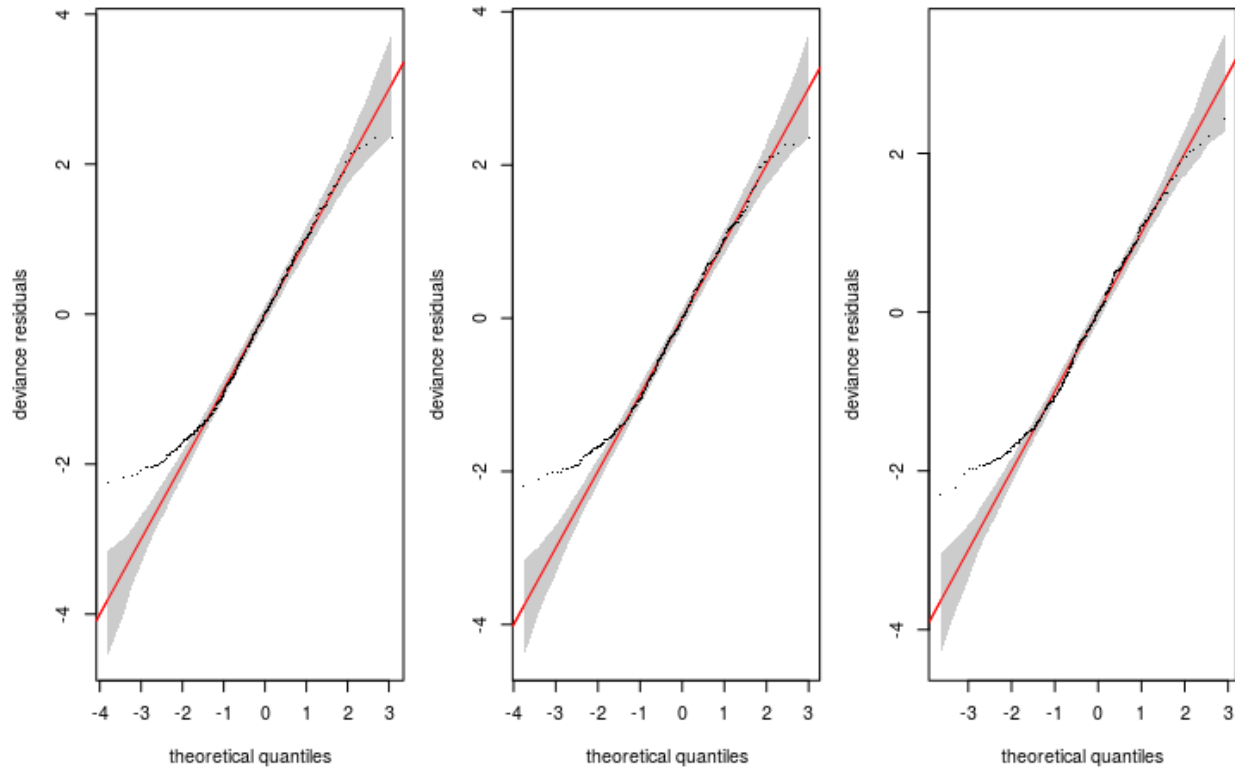


Figure 3: `qq.gam(zoo_daph_mod<1|2|3>, rep = 500)`

model fit.” I don’t agree with the authors’ assessment here – on the contrary, the residual plots show that there are systematic deviations between the observed and expected residual distributions – the lower tail of observed residuals is much longer than the expected one for all 3 models (see figure below).

p. 36: “or using specialized functional regression software such as the `refund` package (Scheipl, Staicu & Greven, 2014), which enforces constraints on the groupwise smoothers so that they always sum to zero at any given point (avoiding the collinearity issue).” The implementation of these “special constraints” in `refund::pffr` is simply using `s(<global-trend-variable>) + ti(<global-trend-variable>, <other-variable>, mc = c(FALSE, TRUE))` – using the rather messy and convoluted (if I say so myself) `refund` package just for that might be overkill, and in any case it should be useful for readers to spell out this possibility explicitly here.

General comments for the author

- please make sure your supplement has the same folder structure and content as your Github repo – e.g. the birds and plankton data sets were missing from the supplement.
- the third author has developed a package for “tidy” ggplots and post-processing of mgcv-fits that looks pretty (!) slick. Not sure why `gratia` is not being used in the code/supplement, that seems like a missed opportunity to me...?

Typos/Language/Graphics:

p. 3, l 81: “github.com/noamross/mixed-effects-gams” should be “github.com/noamross/mixed-effect-gams”

p. 4, l 107: “effect the shape” should be “affect the shape”

p. 5, l 123: “the inverse of the covariance matrix”: more precisely, “the inverse of the (scaled covariance | correlation) matrix” – the smoothing parameter affects this as well, after all.

p. 6, l. 182: “average average”

p. 11, caption Fig 5: “line linetype” should be “linetype”

p. 12, l. 291: “default k=10” - should be **k=10** ...

Figs. 9, 12b: lots of overplotting – please use `geom_point(alpha = .1)` or hexbins.

Fig. 13: shouldn't the plant-specific smooths all use the same `ylim` to show the diversity of the plant-specific trends in shape? Much harder to make out how different they really are with different y-axis limits.

p. 23: “For instance, while table [1]A indicates that models 4 and 5 (which do not have a global function) fit the CO2 data slightly better than models with a global function; however, it is the shape of the mean function that we are actually interested in here, as models 4 and 5 cannot be used to predict the concentration-uptake relationship for other plants.” Use either “while” or “; however”, not both. “shape of the mean function” could be phrased more precisely as “shape of the global mean function”, “other plants” could be phrased more precisely as “plants that are not part of the training data set.”

p. 24: “the area of the net tow and rounded to 1000 give estimated population density per m² for each taxon at each point in time in each sampled lake.” should be “the area of the net tow and rounded to 1000 *to* give estimated population density per m² for each taxon at each point in time in each sampled lake.”

p. 24: “could could borrow information”

p 25: code blocks for models 4/5 appear above the place you wanted them, I think (l. 508).

p.26: “?check.k” should be “?choose.k”, the former gives an error. The actual explanation of the k-index “test” Wood uses here is in `?gam.check`, though.

p. 36: “function `bam()`”

p. 40: “the the unit”

p. 41: “intial idea”

throughout: references to Tables are rendered as “??”

```
devtools::session_info("mgcv")
#Session info -----
# setting  value
# version  R version 3.4.4 (2018-03-15)
# system   x86_64, linux-gnu
# ui       RStudio (1.1.447)
# language en
# collate  en_US.UTF-8
# tz       Europe/Berlin
# date     2018-11-12
#
#Packages -----
# package * version date      source
# graphics * 3.4.4   2018-03-16 local
# grDevices * 3.4.4   2018-03-16 local
# grid      3.4.4   2018-03-16 local
# lattice   0.20-35 2017-03-25 CRAN (R 3.3.3)
# Matrix    * 1.2-14  2018-04-09 CRAN (R 3.4.4)
# methods  * 3.4.4   2018-03-16 local
# mgcv      * 1.8-24  2018-06-18 CRAN (R 3.4.4)
```

```
# nlme      * 3.1-137 2018-04-07 CRAN (R 3.4.4)
# stats     * 3.4.4   2018-03-16 local
# utils     * 3.4.4   2018-03-16 local
```