StatR 502 Homework 6

Gregor Thomas

Due Thursday, Feb 18, 2016, 6:30 pm

Submission guidelines: please submit a knitted PDF or Word document, and optionally your `.Rmd` file. As always, ask in the discussion forum if you're having trouble!

1. Smoothing One Predictor

Using the `LakeHuron` data in the `datasets` package (it should be loaded by default), construct a data frame with two columns, one of lake elevation and one for year. You will want to coerce the the `LakeHuron` data to class `numeric`, the built-in data is a special `ts` time-series class.

Produce a single plot that shows the raw data along with two smoothed lines: one fit by Loess, the other fit with cubic regression splines (`s(..., bs = "cr")` in the `mgcv` package). Make sure both smooths are "tuned" to capture the important-seeming features of the data.

(*Note:* it is possible to do this by fitting the models directly and using the `predict()` method, or by using `geom_smooth` to do the fitting. Either method is fine.)

2. A Real GAM

Use the `mack` data in the `gamair` package. The response variable, `egg.count`, is the count of mackerel eggs in net that has been pulled up to the sea surface from an adequate depth. Many samples were taken, and the latitude/longitude are recorded for each observation, along with quite a few other variables. See `?gamair::mack` for more detail. These data would be of interest for stock assessment of mackerel.

(a) Plot the egg density, `egg.dens`, using longitude/latitude as x/y, and a different aesthetic for the density. (See below for optional addition.)

(b) Despite using `egg.dens` for the plot, for our model we will `egg.count` as the response. Since it's a count, we'll use a Poisson-family GAM. Much like the doctor complain data where we used an offset term because we expected a one-to-one relationship between the number of complaints and the number of hours worked, here we would expect a one-to-one relationship between the number of eggs and the size (cross-sectional area) of net used. Thus, add `+ offset(log(net.area))` to your model formula.

- Use a two-dimensional smooth of longitude and latitude (`s(lon, lat)`) and one-dimensional smooths of depth, distance to the continental shelf, salinity, surface temperature, temperature at 20 m depth, and time.
- *For the 1-d smooths* the default `k` value is probably fine; for the 2-d smooth you should experiment with several different `k` values, assessing the resulting plot. However, don't go *too* high with `k` - I tried `k = 500` and it kept my computer busy for about 10 minutes (it did eventually produce pretty plot, though probably over-fit).
- *For your basis functions,* `mgcv` offers "shrinkage" options as well. We talked a little bit about shrinkage in Lecture 5 - pulling a regression coefficient towards 0. Instead of the the vanilla cubic spline `bs = "cr"` argument, use a cubic spline with shrinkage basis by specifying `bs = "cs"`. Do this for all the smooths *except:* (1) the cubic spline basis with shrinkage does not work for multivariate smooths. Luckily, the thin plate spline basis with shrinkage does, so for the lat/long smooth use `bs = "ts"`. (2) For time of day, it would be really cool to use a *cyclic* smooth. This will make it so that the endpoints of the data (i.e., 11:59 PM and 12:00 AM) have the same smoothness constraints as any other adjacent

data points. To use a cyclic cubic basis, specify `bs = "cc"`. In all cases, the choice of basis `bs` is an argument to the smooth function `s()`.

- *Two more settings:* The `gam()` function has a different way of allowing for overdispersion (quasipoisson) than the `glm` function. Set `scale = -1` to tell `gam` to estimate an overdispersion parameter, otherwise it will act under the assumption that the residual variance is exactly equal to the fitted value. The `gam` function also has an argument, `gamma`, which is a multiplier of the "effective degrees of freedom" in its internal cross validation. The default value of `gamma` is 1, but the Simon Wood book frequently recommends setting `gamma = 1.4` to penalize extra parameters a little bit more and guard against overfitting.
- *Finally!* Fit your model as described *at length* above, and show the model summary and the plots of the smooths produced by `plot(your_model)`. (If you'd like, you can add residuals to these plots by specifying `residuals = TRUE`; see `?plot.gam` for more options).

(c) Several terms should be clearly not helping your model. You can run an `anova` on your model or simply assess the graphic output to drop the un-helpful covariates. Fit a simpler model, and look again at the plots of the smooths.

(d) In one or two sentences, how would you describe the effect of temperature 20 m below the surface on the count of mackerel eggs?

3. Roxygen documentation

(a) Using appropriate text formatting, write a Roxygen-style comment block to document the function you wrote for the basketball simulation on homework 5.

(b) Write a (relatively) short roxygen comment block for either (1) the abalone data set that has been used on the last two homeworks or (2) the data you intend to use for your final project. An example of dataset documentation can be found in the Generating Rd Files vignette. Don't forget the single line of code that must follow the comment block! Note that, to put it in a package, the file with the roxygen data documentation would still be saved as a `.R` file in the `R/` directory within the package.

For this problem, **only turn in the code**—that is the comment blocks themselves. However, create a little package (like in lab 5), put your comment blocks in it (along with your function), run `document()` on it, and view the generated .Rd file to make sure it worked.