GAM modelling workshop: computer lab exercises

Matteo Fasiolo

Software installation

The main packages we will need are `mgcv`, `mgcViz` and `testGam` which should have been installed by doing

```
install.packages("devtools")
library(devtools)
install_github("mfasiolo/testGam")
```

If this worked, stop reading this section and go to the exercises. Otherwise, if `install_github` fails saying "Rate limit remaining: 0/60", then Plan B consists in running the code:

```
install.packages(c("mgcViz", "gamair", "e1071", "languageR",
                   "gamlss.data", "gridExtra", "hexbin"))
```

and then

```
# I will pass you these packages via a USB stick
install.packages("testGam_0.0.1.tar.gz", repos = NULL, type = "source")
```

1 GAMLSS modelling of aggregate UK electricity demand

Here we consider a UK electricity demand dataset, taken from the National Grid. The dataset covers the period January 2011 to June 2016 and it contains the following variables:

- `NetDemand` net electricity demand between 11:30am and 12am.
- `wM` instantaneous temperature, averaged over several cities.
- `wM_s95` exponential smooth of `wM`, that is `wM_s95[i] = a*wM[i] + (1-a)*wM_s95[i]` with `a=0.95`.
- `Posan` periodic index in `[0, 1]` indicating the position along the year.
- `Dow` factor variable indicating the day of the week.
- `Trend` progressive counter, useful for defining the long term trend.
- `NetDemand.48` lagged version of `NetDemand`, that is `NetDemand.48[i] = NetDemand[i-2]`.
- `Holy` binary variable indicating holidays.
- `Year` and `Date` should obvious, and partially redundant.

Questions:

1. Load `mgcViz` and the data (`data("UKload")`). Then create a model formula (e.g. `y~s(x)`) containing: smooth effects for `wM`, `wM_s95` and `Trend` with 20, 20 and 4 knots and cubic regression splines bases (`bs='cr'`), a cyclic effect (`bs='cc'`) for `Posan` with 30 knots; and parametric fixed effects for `Dow`, `NetDemand.48` and `Holy`. Fit a Gaussian GAM using `gamV` with this model formula, and set argument `aViz=list(nsim = 50)` to have some simulated responses for residuals checks. Let `fit0` be the fitted model.

2. Use the `check1D` function together with the `l_gridCheck1D` layer to check whether the conditional mean of the residuals of `fit0` varies along `wM`, `wM_s95` or `Posan`. In the call to `l_gridCheck1D` you can set `stand = "sc"` to standardize the residuals means, thus making the residuals patterns more visible. Look at the plot for `Posan`, does the residuals mean in January (Posan ≈ 0) differ from that in December (Posan ≈ 1)? What does this suggest?

3. Change the model formula, by using a cubic regression spline basis also for `Posan`, and refit the model. Is there any improvement in AIC? Re-check the residuals along `Posan` using `check1D` and `l_gridCheck1D`. Is the pattern gone? Now use `check(fit1)` and look at the p-values. Recall that a low p-value means that an effect might not have a sufficiently large basis. Also, plot all the smooth effects using `plot(fit1)`, how does the effect of `Posan` look like? Given this plot and the result of `check` can you think of a better spline basis for `Posan`?

4. Change the model formula, by using an adaptive spline basis (`bs = 'ad'`) for `Posan`, and refit the model. Is there any improvement in AIC? Now that we are satisfied with our mean model, we start looking at the conditional variance. Use `l_gridCheck1D` with `gridFun = sd` to check for non-constant residuals variance along the same variables. Does the variance change along `wM`, `wM_s95` and `Posan`?

5. Now we will fit a GAMLSS model using the `gaulss` family (see `?gaulss`). For the location use the same model formula we have used in the Gaussian GAM, while for the scale use two cubic regression spline smooths for `wM_s95` and `Posan` (10 and 20 knots respectively) and a fixed effect for `Dow`. Fit the model using `gamV` and then check whether there has been any improvement in AIC, and check the conditional variance again using `l_gridCheck1D`. Is the variance pattern as strong as before? Plot the fitted effects using `plot`.

6. **Extra question**: now that we have a satisfactory model for the conditional variance, we look at further features of the residuals distribution. Plot a QQ-plot of the residuals of `fit3` using `qq`. Do you see significant deviations from the model-based theoretical residuals distribution? Load the `e1071` package and use `check1D` with `l_gridCheck1D` and `gridFun = skewness` to verify how the skewness of the residuals varies along `wM_s95` and `Posan`. Do you see major departures from the model-based simulations?

7. **Extra question**: to allow the distribution of the response to be skewed we will now consider the `shash` distribution (see `?shash`). The `shash` family has four parameters, so we need to specify four linear predictors (location, scale, skewness and kurtosis in that order) in the model formula. For location and scale use the same models we used for `gaulss`, for the skewness include a fixed (linear) effect for `Dow` and a smooth effect for `Posan` (with `k = 10` and `bs='cr'`), while for the kurtosis use only an intercept (`~ 1`). Fit the model, convert it and call it `fit4`. Check whether the AIC has improved, relative to `fit3` and produce another QQ-plot using `qq`. Are the deviations from the theoretical distribution larger or smaller in this model?

8. **Extra question**: well... congratulations if you got here! What one could do at this point is to check how the kurtosis changes along the covariates using `l_gridCheck1D` (`e1071` provides a

function called `kurtosis`). But beware: the `shash` might break down during model fitting if you try to fit overly complicated models.