Second session: computer lab exercises

In this session you could try one or more of the following exercises on electricity forecasting:

1. GAMLSS modelling of aggregate UK electricity demand (solution in "UKload_GAMLSS.html"). Interactive model-building on UK aggregate electricity demand.

2. Quantile modelling of UK electricity demand (sol: "UKload_QGAM.html"). Similar to previous exercise, but using quantile GAMs.

3. Solar production modelling (sol: "solar_production.html"). We compare QGAM and GAMLSS model for predicting aggregated solar production from 300 installations in Sidney.

Otherwise you could try one of these other exercises, not focused on the electricity industry:

4. GAMLSS modelling Body Mass Index (BMI) of Dutch boys (sol: "bmi_GAMLSS.html"). Basic exercise featuring adaptive smoothers.

5. GAMLSS rent modelling in Munich (sol: "Rent_munich_GAMLSS.html"). Featuring linear interactions.

6. QGAM modelling of rainfall in Switzerland (sol: "Swiss_rainfall_QGAM.html"). Featuring spatio-temporal effects constructed using tensor product bases.

but feel free to try `qgam` and `mgcViz` on your own data.

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 English cities.
- `wM_s95` exponential smooth of `wM`, that is `wM_s95[i] = a*wM[i] + (1-a)*wM_s95[i-1]` 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-1]`.
- `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 `check1D` together with the `l_densCheck` layer to compare the empirical and theoretical (Gaussian) density of the residuals along `wM`, `wM_s95` an `Posan`. Do you see any evidence of model mis-specification? Now 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 from the `mgcFam` package (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 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` is still experimental and model estimation might break down if you try to fit overly complicated models.

2 Quantile modelling of 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 English cities.
- `wM_s95` exponential smooth of `wM`, that is `wM_s95[i] = a*wM[i] + (1-a)*wM_s95[i-1]` 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-1]`.
- `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`, `Posan` and `Trend` with 20, 20, 50 and 4 knots and cubic regression splines bases (`bs='cr'`); parametric effects for `Dow`, `NetDemand.48` and `Holy`.

2. Use the `qgamV` function to fit this model for the median. Call (say) `fit` the fitted model and use `plot(fit)` and `summary(fit)` to visualise the fitted effects and to see which effects are significant. Do you notice anything problematic about the effect of `Posan`? How many degrees of freedom are we using for this smooth effect (you can read it from the output of `summary`)?

3. Modify the effect of `Posan` to use an adaptive (`bs='ad'`) spline basis. Then refit the model and plot the smooth effects. Has the effect of `Posan` changed? How many degrees of freedom are we using now for `Posan`? Explain what happened.

4. Use `mqgamV` to fit this model to the five quantiles `qu=seq(0.1,0.9,length.out=5)`. Use `plot` to visualize the smooth effects corresponding to each quantile. You can set `allTerms=TRUE` to plot also the parametric effects. How do the smooth and parametrics effects differ between quantiles? NB: here we are plotting the smooth effects, not the predicted quantiles, hence the effects corresponding to, say, quantile 0.9 can fall below that of quantile 0.1.

5. Now we check the median fit. If the output of `mqgamV` is called `fitM` then the median fit is `fitM[[3]]`. Use `check1D` with the `l_gridQCheck1D` layer to check that the fraction of negative residuals does not depart too much from 0.5 along any of the covariates.

3 Solar production modelling

Here we have data on aggregate solar electricity production from residential solar panels installed in 300 locations around Sidney. The raw data is here: https://www.ausgrid.com.au/Common/About-us/Corporate-information/ Data-to-share/Solar-home-electricity-data.aspx. We want to model production using time-of-day and time-of-year effects, and we compare QGAM and GAMLSS models in terms of predictive performance. The dataset contains the following variables:

- `prod` total production in a 30min time slot;
- `Posan` periodic index in `[0, 1]` indicating the position along the year;
- `Instant` the time of day, where 0 corresponds to 00:00-00:30, 1 to 00:30-01:00 and so on;
- `date` date and time;
- `dow` the day of the week;
- `logprod` this is $\log(\text{prod} + 0.01)$;

Questions:

1. Load `testGam`, `mgcViz` and the data (`data("solar_prod")`). Divide the data into a training and a testing set, by doing:

   ```
   set.seed(515)
   iTest <- sample(1:nrow(solar_prod), 2000)
   DataTEST <- solar_prod[iTest, ]
   DataTRAIN <- solar_prod[-iTest, ]
   ```

   Admittedly when forecasting on the test set we will be intrapolating (so this is not very realistic), but we have only on year of data.

2. Fit a quantile GAM for the median production using `qgamV` with a fixed effect for `dow` and a tensor product smooth for the joint effect of `Instant` and `Posan`. For the latter use cyclical bases (`bs=c("cc","cc")`) and `k=c(5,5)`. Plot the 2D effect interactively using `plotRGL` (with `residuals=TRUE`). Do you see any anomalous residual pattern along `Instant`?

3. Use `check2D` and the pinball loss function:

   ```
   # y = observed logprod, mu = predicted quantile, qu is quantile of interest
   pinball <- function(y, mu, qu){
     tau <- 1 - qu
     d <- y - mu
     l <- d * 0
     l[d < 0] <- - tau*d[d<0]
     l[d > 0] <- - (tau-1)*d[d>0]
     return( l )
   }
   ```

   to check how the loss changes along `Instant` and `Posan`. To bin the losses in 2D you can add the `l_gridCheck2D(mean, stand = F)` layer.

4. Increase the number of basis functions used for `te(Posan, Instant)` to `k=c(5,15)`, re-fit and repeat the pinball loss checks. Any improvement? Use also `l_gridQCheck2D()` to verify whether the proportion of negative residuals varies along `Instant` and `Posan`.

5. Use `mqgamV` to fit the same quantile GAM to the quantiles `qus=seq(0.1,0.9,length.out = 5)`. Let's assume that the output of `mqgamV` is called fit `fitMQ`. Use the following code:

```
fit50 <- getViz(fitMQ[[3]], newdata = DataTEST)

pl <- check1D(fit50, list("Instant", "Posan"), type = "y",
              trans = function(x, ...) pinball(x, predict(fit50, newdata = DataTEST), qu = 0.5)) +
  geom_smooth() +
  l_gridCheck1D(gridFun = mean, level = 0) +
  l_rug() + ylab("Pinball loss")

print(pl, pages = 1)
```

to visualise how the pinball loss changes with `Instant` and `Posan`, for quantile 0.5 and on the test set. Is the loss higher during any specific time of the day or of the year? If so, why do you think that's the case?

6. Now we consider a GAMLSS approach. Use `gamV` to fit a Gaussian GAMLSS model (`family = gaulss`), with model the same mean model as we used for the median. For the variance you can use the same model, but set `k=c(5,5)` for the tensor effect. Plot the tensor effect of `Instant` and `Posan` and try to interpret its shape.

7. Now we compare QGAM and GAMLSS in terms of predictive performance. Fit a Gaussian GAM with the same model formula as the quantile GAM, and define the following function (you should be able to copy-paste it):

```
compareMod <- function(QGAM, GAUS, GAULSS, dat, variab){

  nq <- length(QGAM)
  qus <- as.numeric(names(QGAM))

  pinQv <- lapply(1:nq, # QGAM
                  function(.ii){
                    tmp <- getViz(QGAM[[.ii]], newdata = dat)
                    check1D(tmp, variab, type = "y",
                            trans = function(x, ...)
                              pinball(x, predict(tmp, newdata = dat), qu = qus[.ii]))
                  })

  pinGAUSv <- lapply(1:nq, # GAUSSIAN GAM
                     function(.ii){
                       tmp <- getViz(GAUS, newdata = dat)
                       check1D(tmp, variab, type = "y",
                               trans = function(x, ...)
                                 pinball(x,
                                         qnorm(qus[.ii], predict(tmp, newdata = dat),
                                               sqrt(fitGAUS$sig2)),
                                         qu = qus[.ii]))
                     })
```

```r
  pinLSSv <- lapply(1:nq, # GAULSS
                    function(.ii){
                      tmp <- getViz(GAULSS, newdata = dat)
                      check1D(tmp, variab, type = "y",
                              trans = function(x, ...){
                                .pred <- predict(tmp, newdata = dat, type = "response")
                                pinball(x, qnorm(qus[.ii], .pred[ , 1], 1 / .pred[ , 2]),
                                        qu = qus[.ii])
                              })
                    })

  plts <- list()
  for(ind in 1:nq){
    plDat <- as.data.frame(rbind(pinQv[[ind]]$data$res, pinGAUSv[[ind]]$data$res,
                                 pinLSSv[[ind]]$data$res))
    plDat$plotID <- as.factor(rep(c("QGAM", "GAU", "GAULSS"), each = nrow(plDat)/3))

    plts[[ind]] <- ggplot(data = plDat,
                          mapping = aes(x = x, y = y, group = plotID,
                                        col = plotID, fill = plotID)) +
      geom_smooth(se = FALSE) + theme_bw() + ggtitle(qus[ind])
  }

  return(plts)

}
```

Then you should be able to compare the performance of the three models (across `Instant` or `Posan`, and all quantiles) by doing:

```r
# I assume that fitMQ is output of mqgam, fitGAUS is Gaussian GAM and fitLSS is gaulss GAM
plts <- compareMod(QGAM = fitMQ, GAUS = fitGAUS, GAULSS = fitLSS, dat = DataTEST,
                   variab = "Instant")
library(gridExtra)
grid.arrange(grobs = plts)
```

Which model does better?

4 Body Mass Index (BMI) of Dutch boys

This simple data set comes from the Fourth Dutch Growth Study, which is a cross-sectional study that measures growth and development of the Dutch population between the ages 0 and 21 years. Here we have only two variables: `bmi` and `age`. The data is taken from the `gamlss.data` package.
Questions:

1. Load `testGam`, `mgcViz` and the data (`data("dbbmi")`). Then use `gamV` to fit a Gaussian GAM with simply a single smooth effect for `age`. Set argument `aViz=list(nsim = 50)` to have some simulated responses for residuals checks. Then plot the data (a scatterplot `bmi` vs `age`) and add a line representing the fitted mean BMI (you can use the `predict` function).

2. Check the residual distribution using `qq`: do you see any problem? Then use the `check1D` function together with the `l_gridCheck1D(gridFun=sd)` layer to check whether the conditional standard deviation of the residuals varies with age. If so, address this by fitting a Gaussian GAMLSS model (`family = gaulss`), with model formula `list(bmi ~ s(age), ~ s(age))`. Then repeat the residuals checks. Any improvement?

3. Use `check` to verify whether the number of basis functions used for the smooth effects is sufficiently large. Then increase the number of basis functions used for each effect to 20 (`k=20`), and use an adaptive basis fom the effect of age on mean BMI (`bs = "ad"`). Is this model better in terms of AIC? Does the output of `check` look ok now? Plot the smooth effects, and decide whether they make sense. Do you see why we used an adaptive smooth for the effect of age on mean BMI?

4. Now we look at residual skewness. Load the `e1071` package, and use the `check1D` function together with the `l_gridCheck1D(gridFun=skewness)` layer to check whether the conditional skewness of the residuals varies with age. To take skewness into account, load the `mgcFam` package, and fit a shash GAM model (`family=shash`) with model formula:

```
list(bmi ~ s(age, k = 20, bs = "ad"), ~ s(age, k = 20), ~ s(age), ~ 1)
```

   Do we get lower AIC, and how does a residuals QQ-plot look? Plot all the smooth effect and use `check` to verify that everything is ok.

5. Now we plot the fitted conditional distribution. Let `fit4` be the shash model you just fitted, then you can plot several estimated conditional quantiles by doing:

```
plot(bmi~age, data=dbbmi, col = "grey")
pr <- predict(fit4)
for(.q in c(0.01, 0.25, 0.5, 0.75, 0.9)){
  q_hat <- fit4$family$qf(.q, pr, wt = fit4$prior.weights, scale = 1)
  lines(dbbmi$age, q_hat, col = 2)
}
```

5 Rent modelling in Munich

This data set comes from `gamlss.data` package. The main variables are:

- `R` rent response variable, the monthly net rent in DM;
- `Fl` floor space in square meters;
- `A` year of construction;
- `B` a binary indicating whether there is a bathroom, 1, (1925 obs.) or not, 0, (44 obs.);
- `H` a binary indicating whether there is central heating, 1, (1580 obs.) or not, 0, (389 obs.);
- `L` a binary indicating whether the kitchen equipment is above average, 1, (161 obs.) or not, 0, (1808 obs.);
- `loc` a factor indicating whether the location is below, 1, average, 2, or above average 3.

Questions:

1. Load testGam, mgcViz and the data (data("munich_rent")) and have a look at it by doing pairs(munich_rent). Then use gamV to fit a Gaussian GAM with rent as response, smooth effects for Fl and A and fixed effects for the remaining covariates. Set argument aViz=list(nsim = 50) to have some simulated responses for residuals checks. Use summary and plot to see which are the most important effects.

2. The effect of Fl looks fairly linear, but it should depend on the location's desirability (loc). Substitute the smooth effect for Fl with a linear effect for Fl and the interaction Fl:loc. Is there any improvement in AIC? Do the fitted coefficient reported by summary make sense?

3. Use the check1D function together with the l_gridCheck1D(gridFun=sd) layer to check whether the conditional standard deviation of the residuals varies with any of the covariates. If so address this by fitting a Gaussian GAMLSS model (family = gaulss), with the same formula for mean and for scale. Then repeat the residuals checks. Any improvement? Do you get lower AIC?

4. Now look at the residuals distribution using qq. Do you see any departure from normality? Check whether the conditional skewness of the residuals varies with any of the covariates by loading the e1071 package, and using the check1D function together with the l_gridCheck1D(gridFun=skewness) layer. To take skewness into account, load the mgcFam package, and fit shash GAM model (family=shash) with model formula:

```
list(R ~ Fl + Fl:loc + s(A) + loc + B + H + L,
       ~ Fl + Fl:loc + s(A) + loc + B + H + L,
       ~ Fl + Fl:loc + s(A) + loc + B + H + L,
       ~ 1)
```

   Do we get lower AIC, and how does a residuals QQ-plot look? Do the skewness checks obtained with check1D look better now? Finally plot the smooth effects.

6 Rainfall modelling in Switzerland

This question is about modelling extreme rainfall in Switzerland, mainly using spatio-temporal effects. The main variables are:

- exra: the highest rainfall observed in any 12 hour period in that year, in mm;
- N: degrees North;
- E: degrees East;
- elevation: metres above sea level;
- climate.region: factor variable indicating one of 12 climate regions;
- nao: annual North Atlantic Oscillation index, based on the difference of normalized sea level pressure (SLP) between Lisbon, Portugal and Stykkisholmur/Reykjavik, Iceland. Positive values are generally associated with wetter and milder weather over Western Europe;
- year: year of the observation;

Questions:

1. Load `mgcViz`, `gamair` and the data with `data(swer)`. Use `qgamV` to fit an additive quantile regression model for the median of `exra`, with smooth effects for `nao`, `elevation` and `year` (use `k=5` for the latter), an isotropic smooth for `E` and `N` (i.e. `s(E,N)`), and a fixed effect for `climate.region`. Look at the significance of the fitted effects using `summary` and plot them using `plot`.

2. We might be interested in verifing whether the rainfall trend is different depending on the climate region. To assess this, modify the model formula to include a by-factor smooth as follows `s(year, climate.region, bs = "fs", k = 5)` (you will have to remove the fixed `climate.region` effect). Refit and use `summary` to verify whether the by-region trend term is significant, and plot the by-region trends by extracting it using `sm` and the `l_fitLine(alpha = 1)` layer.

3. We can also verify whether the bivariate spatial effect changes with time, by creating a tensor product between the 2D effect of `E` and `N`, and the effect of `year`. Such an effect can be set up using `te(E, N, year, d = c(2, 1), k = c(20, 5))`. Fit the corresponding median QGAM model, and plot several slices of the 3D tensor product across `year`, using the `plotSlice` function with the `l_fitRaster` and `l_fitContour` layers.

4. Visualize individual 2D slices (across `year`) of the 3D spatio-temporal smooth using the `plotRGL` function (see `?plotRGL.mgcv.smooth.MD` for examples).

5. Go back to the simpler model formula used in the first question and fit the corresponding model to the quantiles `qu = seq(0.1, 0.9, length.out = 9)`, using `mqgamV`. Plot only the univariate effects using `plot` and its `select` argument, and see how they differ between quantiles. Do the same for the spatial effect and for the effect of the climate region.