Weekly Assignment 9: Splines and GAMs

Marjolein Fokkema

Context and dataset

In this exercise, we will predict to what extent the daily number of cyclists on four bridges in New York is affected by temperature, precipitation, weekday and month. We use a dataset with daily cyclist counts obtained over several months in 2006. Get the file "bike_dat.Rdata" from Brightspace, and load it into `R` as follows:

```
bike_dat <- readRDS("bike_dat.Rdata")
```

The dataset contains the following variables:

- `Date`: Factor indicator for the date.
- `Day`: Factor indicator for day of the week.
- `High.Temp`: Numeric indicator for highest temperature of the day.
- `Low.Temp`: Numeric indicator for lowest temperature of the day.
- `Total`: The total number of cyclists counted on all New York bridges (not used).
- `snow`: Binary indicator for whether it snowed that day.
- `prec`: Numeric indicator reflecting the amount of precipitation that day.
- `rain`: Binary indicator for whether it snowed that day.
- `day`: Numeric indicator for day of the month (not used).
- `month`: Factor indicator for month.
- `counts`: The response variable of interest, count of the number of cyclists on the bridge (indicated by `bridge`) that day.
- `bridge`: Factor indicating on which bridge the count of cyclists was obtained.

Exercise

Use your student number to set the random seed, and set apart a random sample of 200 observations to test the predictive performance of your model(s).

a) Fit a GAM using function `gam()` from package `mgcv` to predict daily cyclist counts on the bridges of New York:

- Specify `counts` as the response.

- As predictor variables, use all remaining variables, except `total` and `day`:
  - For categorical / factor variables, you cannot fit a spline, so you include them as predictors as you would normally do in a GLM: simply specify them in the model formula.
  - For the numeric predictors, you wrap them in the smoothing spline function `s()`. Note that function `s()` (from package `mgcv`) has two main arguments in addition to the predictor variable specified:
    * `k`, which specifies the number of basis functions used for the spline (the default is 9 basis functions).
    * `bs`, which specifies the type of spline function. The default is `bs = "tp"`, yielding a thin-plate spline basis. A range of other options are available, e.g., by specifying `bs = "cr"` yields a cubic spline basis. By specifying `bs = "re"` and a factor variable as the predictor, we obtain a random effect estimated with respect to the factor variable.
  - Include a random intercept w.r.t. `Date` in the model: `s(Date, bs = "re")`. It is likely that measurements made on the same day will be correlated, including a random intercept term for `Date` will account for this.
- Fit the GAM using only the training observations.
- Make sure to use REML estimation by specifying `method = "REML"` in the call to function `gam()`

b) Interpret the fitted model using the `summary` and `plot` methods:

- In which month and on which weekday are New York bridges most (least) busy with cyclists? (You can gather this from the parametric part of the output of `summary`).
- Do snow days yield different cyclist counts than dry days? Than rain days?
- Is the effect of temperature and precipitation significant? Use the output of the `summary` function under `Approximate significance of smooth terms` to evaluate this.
- Describe the effect of temperature and precipitation based on the fitted smoothing spline curves obtained with `plot()`. Evaluate the strength of non-linearity using the `edf` values from the output of `summary` (`edf` of 1 corresponds to a linear effect, higher values indicate stronger non-linearity).
- Evaluate the distribution of the random intercept estimated w.r.t. `Date` from the plot. Are there large differences in counts? Do the counts follow a normal distribution?

c) Evaluate predictive accuracy (MSE) of the fitted GAM on the test observations.