Lab 3.4: Generalized Linear Spatio-Temporal Regression

In this Lab we fit a generalized linear spatio-temporal model to yearly counts of Carolina wren in and around the state of Missouri between 1994 and 2014. We need **gstat**, **sp**, and **spacetime** for fitting an empirical semivariogram to the residuals, **FRK** to construct the basis functions (as in Lab 3.2), **ape** for running Moran's I test, and the usual packages for wrangling and plotting.

```
library("ape")
library("dplyr")
library("FRK")
library("ggplot2")
library("gstat")
library("sp")
library("spacetime")
library("STRbook")
library("tidyr")
```

Fitting the Model

The Carolina wren counts in the BBS data set, in both wide and long format, are supplied with **STRbook**. Here we load the data directly in long format and remove any records that contain missing observations.

```
data("MOcarolinawren_long", package = "STRbook")
MOcarolinawren_long <- MOcarolinawren_long %>%
                       filter(!is.na(cnt))
```

We use the same covariates to fit these data as we did to fit the maximum temperature, `Tmax`, in Lab 3.2. Twelve of these covariates were basis functions constructed using `auto_basis` from the package **FRK**; see Lab 3.2 for details. The matrix `S` below then contains the basis functions evaluated at the Carolina wren observation locations.

```
G <- auto_basis(data = MOcarolinawren_long[,c("lon","lat")] %>%
                       SpatialPoints(),           # To sp obj
                nres = 1,                         # One resolution
                type = "Gaussian")                # Gaussian BFs

S <- eval_basis(basis = G,                       # basis functions
                s = MOcarolinawren_long[,c("lon","lat")] %>%
                     as.matrix()) %>%            # conv. to matrix
     as.matrix()                                 # conv. to matrix
colnames(S) <- paste0("B", 1:ncol(S)) # assign column names
```

Next, we attach the basis-function covariate information to the data frame containing the counts, and remove the fields `loc.ID` and `t`, which we will not explicitly use when fitting the model. We list the first five columns of the first three records of our constructed data frame `Wren_df` as follows.

```
Wren_df <- cbind(MOcarolinawren_long,S) %>%
  select(-loc.ID, -t)
Wren_df[1:3, 1:5]

##   cnt  lat   lon year      B1
## 1   4 36.8 -89.2 1994 0.00258
## 2   2 36.6 -90.7 1994 0.03551
## 3   8 36.9 -91.7 1994 0.11588
```

Generalized linear models (GLMs) are fitted in `R` using the function **`glm`**. The function works similarly to **`lm`**, but in addition it requires one to specify the exponential-family model that is used (in this first instance we consider the Poisson family), as well as the link function (here we use the log function, which is the canonical link). The **`glm`** function is called as follows (note that we have used the same formula as in Lab 3.2).

```
Wren_GLM <- glm(cnt ~ (lon + lat + year)^2 + ., # formula
                family = poisson("log"),      # Poisson + log link
                data = Wren_df)               # data set
```

The mean and variance of a random variable that has a Poisson distribution are the same. In cases where the variance in the data is greater than that suggested by this model, the data are said to exhibit "over-dispersion." An estimate of the dispersion is given by the ratio of the deviance to the total degrees of freedom (the number of data points minus the number of covariates). In this case the dispersion estimate is

```
Wren_GLM$deviance / Wren_GLM$df.residual

## [1] 3.78
```

which is greater than 1, a sign of over-dispersion.

Another way to obtain an estimate of the disperson parameter (and, to account for it if present) is to replace **`poisson`** with **`quasipoisson`** when calling **`glm`**, and then type **`summary`**`(Wren_GLM)`. The quasi-Poisson model assumes that the variance is proportional to the mean, and that the constant of the proportionality is the over-dispersion parameter. Note from the output of **`summary`** that the dispersion parameter is `3.9`, which is close to what we estimated above.

It can be shown that under the null hypothesis of no over-dispersion, the deviance is approximately chi-squared distributed with degrees of freedom equal to $m - p - 1$.

```
Wren_GLM$df.residual

## [1] 764
```

The observed deviance is

```
Wren_GLM$deviance

## [1] 2890
```

The probability of observing such a large or larger deviance under the null hypothesis of no over-dispersion (i.e., the p-value) is

```
1 - pchisq(q = Wren_GLM$deviance, df = Wren_GLM$df.residual)

## [1] 0
```

Therefore, we reject the null hypothesis of no over-dispersion at the usual levels of significance (10%, 5%, and 1%). One may use other models in the exponential family, such as the negative-binomial distribution, to account explicitly for the over-dispersion. For convenience, in this Lab we proceed with the Poisson family.

Prediction

As in the other Labs, prediction proceeds through use of the function `predict`. We first generate our space-time prediction grid, which is an 80 × 80 × 21 grid in degrees × degrees × years, covering the observations in space and in time.

```
pred_grid <- expand.grid(lon = seq(
                             min(MOcarolinawren_long$lon) - 0.2,
                             max(MOcarolinawren_long$lon) + 0.2,
                             length.out = 80),
                         lat = seq(
                             min(MOcarolinawren_long$lat) - 0.2,
                             max(MOcarolinawren_long$lat) + 0.2,
                             length.out = 80),
                         year = 1994:2014)
```

As in Lab 3.2, we now evaluate the basis functions at the prediction locations.

```
S_pred <- eval_basis(basis = G,                     # basis functs
                s = pred_grid[,c("lon","lat")] %>%  # pred locs
                     as.matrix()) %>%               # conv. to matrix
     as.matrix()                                    # as matrix
colnames(S_pred) <- paste0("B", 1:ncol(S_pred))   # assign  names
pred_grid <- cbind(pred_grid,S_pred)              # attach to grid
```

In the call to `predict` below, we specify `type = "link"` to indicate that we predict the link function of the response and not the response (analogous to the log-intensity of the process).

```
wren_preds <- predict(Wren_GLM,
                      newdata = pred_grid,
                      type = "link",
                      se.fit = TRUE)
```

The predictions and prediction standard errors of the link function of the response are then attached to our prediction grid for plotting. Plotting is left as an exercise for the reader.

```
pred_grid <- pred_grid %>%
             mutate(log_cnt = wren_preds$fit,
                    se = wren_preds$se.fit)
```

When fitting GLMs, it is good practice to check the deviance residuals and inspect them for any residual correlation. The default GLM residuals returned by `residuals` are deviance residuals.

```
Wren_df$residuals <- residuals(Wren_GLM)
```

Interestingly, the plot of the deviance residuals in Figure 1 is "noisy," indicating a lack of spatial correlation.

```
g2 <- ggplot(Wren_df) +
    geom_point(aes(lon, lat, colour = residuals)) +
    col_scale(name = "residuals") +
    facet_wrap(~year, nrow = 3) + theme_bw()
```

We can test for spatial correlation of the deviance residuals by running Moran's *I* test on the spatial deviance residuals for each year. The code below follows closely that for Moran's *I* test in Lab 3.2 and then summarizes the *p*-values obtained for each year.

```
P <- list()                                 # init list
years <- 1994:2014
for(i in seq_along(years)) {                # for each day
  Wren_year <- filter(Wren_df,
                     year == years[i])      # filter by year
  obs_dists <- Wren_year %>%                # take the data
    select(lon,lat) %>%                     # extract coords.
    dist() %>%                              # comp. dists.
    as.matrix()                             # conv. to matrix
  obs_dists.inv <- 1/obs_dists              # weight matrix
```

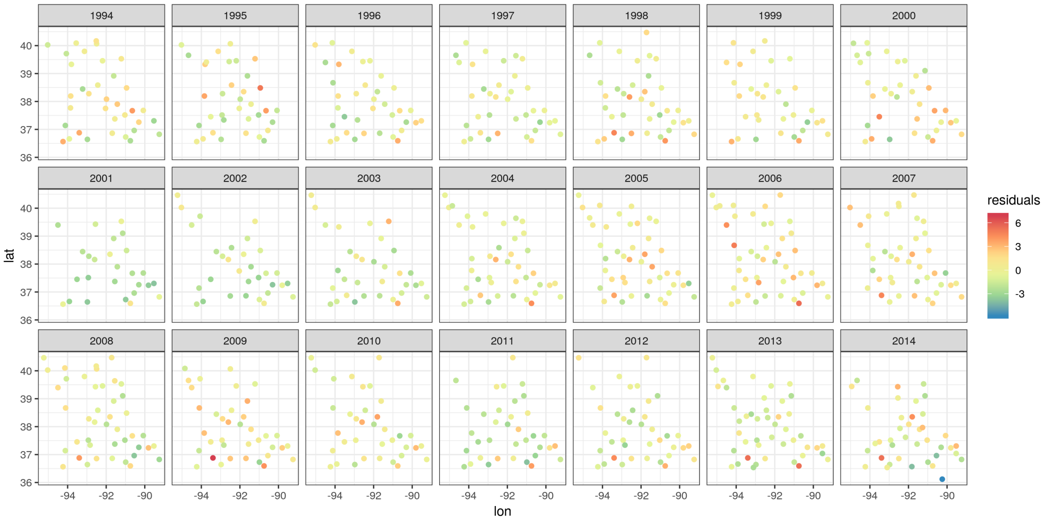

Figure 1: The deviance residuals from the fitted GLM between $t = 1$ (the year 1994) and $t = 21$ (2014).

```
  diag(obs_dists.inv) <- 0                   # 0 on diag
  P[[i]] <- Moran.I(Wren_year$residuals,     # run Moran's I
                    obs_dists.inv) %>%
            do.call("cbind", .)              # conv. to df
}
do.call("rbind",P) %>% summary(digits = 2)

##     observed         expected             sd           p.value
##  Min.   :-0.084   Min.   :-0.040   Min.   :0.025   Min.   :0.06
##  1st Qu.:-0.059   1st Qu.:-0.029   1st Qu.:0.028   1st Qu.:0.24
##  Median :-0.044   Median :-0.029   Median :0.030   Median :0.42
##  Mean   :-0.041   Mean   :-0.028   Mean   :0.031   Mean   :0.47
##  3rd Qu.:-0.022   3rd Qu.:-0.025   3rd Qu.:0.033   3rd Qu.:0.68
##  Max.   : 0.010   Max.   :-0.023   Max.   :0.041   Max.   :0.94
```

Hence, at the 5% level of significance, the null hypothesis (of no spatial correlation in these deviance residuals) is not rejected. This was expected from the visualization in Figure 1.

More insight can be obtained by looking at the empirical semivariogram of the deviance residuals. To do this we first construct an `STIDF`, thereby casting the irregular space-time data into a **spacetime** object.

```
Wren_STIDF <- STIDF(sp = SpatialPoints(
                            Wren_df[,c("lon","lat")],
                            proj4string = CRS("+proj=longlat")),
                    time = as.Date(Wren_df[, "year"] %>%
                                       as.character(),
                                   format = "%Y"),
                    data = Wren_df)
```

Then we compute the empirical semivariogram using **variogram**. We consider time bins of width 1 year (i.e., of width 52.1429 weeks). Bins specified in units of weeks are required, as this is the largest temporal unit recognized by **variogram** .

```
tlags <- seq(0.01, 52.1429*6 + 0.01, by = 52.1429)
vv <- variogram(object = residuals ~ 1, # fixed effect component
                data = Wren_STIDF,      # data set
                tlags = tlags,          # temp. bins
                width = 25,             # spatial bin (25 km)
                cutoff = 150,           # use pts < 150 km apart
                tunit = "weeks")        # time unit
```

The empirical semivariogram can be plotted using **plot**(vv). Notice how there is little evidence of spatial correlation but ample evidence of temporal correlation in the residuals. (The variance of the differences over a large range of time lags at the same spatial location is small.) This is a clear sign that a more sophisticated spatio-temporal random-effects model should be considered for these data.