Advanced Bayesian Methods: **Theory and Applications in R**

Discrete Spatial Smoothing

Nikolaus Umlauf
https://nikum.org/abm.html

Spatial Effects for Regional Data

- Each observation i is assumed to belong to one of the spatial regions represented by a spatial indicator $r_i \in \{1, \ldots, L\}$.
- Assign separate regression coefficients γ_l, $l = 1, \ldots, L$, to each of the L regions.
- The spatial effect $\gamma_{r_i} = f(r_i)$ of an individual observation i collected in region r_i can then be expressed as

$$f(r_i) = \sum_{l=1}^{L} \gamma_l B_l(r_i),$$

where

$$B_l(r_i) = \begin{cases} 1 & \text{if } r_i = l \\ 0 & \text{otherwise.} \end{cases}$$

Spatial Effects for Regional Data

- In matrix notation this yields the $(n \times L)$ design matrix $\boldsymbol{B}$ with entries

$$\boldsymbol{B}[i,l] = \begin{cases} 1 & \text{if } r_i = l \\ 0 & \text{otherwise} \end{cases}$$

  and the complete vector of spatial effects is given by $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_L)'$.
- Assume a Gaussian Markov random field prior, where the conditional distribution of γ_l given all the neighboring effects is specified as

$$\gamma_l \,|\, \gamma_r, r \neq l \sim \mathrm{N}\left(\frac{1}{|N(l)|}\sum_{r:r\sim l}\gamma_r, \frac{\tau^2}{|N(l)|}\right),$$

  where $l \sim r$ indicates that l and r are neighbors and $|N(l)|$ is the number of neighbors of region l.

Spatial Effects for Regional Data

- This implies that
  - the prior expectation for the spatial effect in region l is given by the average of all spatial effects of neighboring regions,
  - the effect in region l is conditionally independent of all non-neighbors, and
  - the variance of the conditional prior distribution in region l is inversely proportional to the number of neighbors.

Spatial Effects for Regional Data

- The conditional distributions yield a multivariate Gaussian joint distribution for γ given by

$$p(\gamma \mid \tau^2) \propto \left(\frac{1}{\tau^2}\right)^{\mathrm{rk}(\boldsymbol{K})/2} \exp\left(-\frac{1}{2\tau^2}\gamma' \boldsymbol{K} \gamma\right),$$

  with prior precision matrix

$$\boldsymbol{K}[l,r] = \begin{cases} -1 & l \neq r, l \sim r, \\ 0 & l \neq r, l \nsim r, \\ |N(l)| & l = r, \end{cases}$$

- If each region has at least one neighbor and the map is fully connected, the rank of the spatial adjacency matrix is given by $\mathrm{rk}(\boldsymbol{K}) = L - 1$.

Belgium Insurance

Example in R:

```
R> library("bamlss")
R> library("sf")
```

Load the data

```
R> d <- load("Belgium/BelgiumInsurance.rda")
R> print(d)
```

```
[1] "BelgiumInsurance" "Belgium"
```

Belgium Insurance

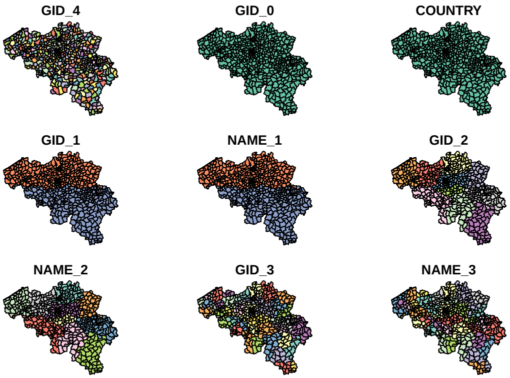
Insurance claim data.

```
R> head(BelgiumInsurance)
  amount ageph agec bm fleet coverage region power    pop     use      lat
1  65270    50   12  5    no        1  21004    77 133138 private 50.85555
2   6292    28    7  9    no        1  21004    70 133138 private 50.85555
3   6292    26    8 11    no        3  21004    55 133138 private 50.85555
4  21980    39    2  1    no        3  21004    85 133138 private 50.85555
5  10167    34    9  6    no        1  21004   119 133138 private 50.85555
6   9383    33    9  4    no        1  21004   101 133138 private 50.85555
       lon              NAME_4    sex nclaims
1 4.369724 Sint-Joost-ten-Node female       1
2 4.369724 Sint-Joost-ten-Node   male       1
3 4.369724 Sint-Joost-ten-Node female       1
4 4.369724 Sint-Joost-ten-Node female       1
5 4.369724 Sint-Joost-ten-Node female       1
6 4.369724 Sint-Joost-ten-Node female       2
```

Belgium Insurance

Plot the map of Belgium.

```
R> plot(Belgium)
```



Belgium Insurance

```
R> print(head(Belgium))
```

```
Simple feature collection with 6 features and 15 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 4.243638 ymin: 50.79935 xmax: 4.435751 ymax: 50.91029
Geodetic CRS:  WGS 84
          GID_4 GID_0 COUNTRY   GID_1    NAME_1     GID_2    NAME_2       GID_3
1 BEL.1.1.1.1_1   BEL Belgium BEL.1_1 Bruxelles BEL.1.1_1 Bruxelles BEL.1.1.1_1
2 BEL.1.1.1.2_1   BEL Belgium BEL.1_1 Bruxelles BEL.1.1_1 Bruxelles BEL.1.1.1_1
3 BEL.1.1.1.3_1   BEL Belgium BEL.1_1 Bruxelles BEL.1.1_1 Bruxelles BEL.1.1.1_1
4 BEL.1.1.1.4_1   BEL Belgium BEL.1_1 Bruxelles BEL.1.1_1 Bruxelles BEL.1.1.1_1
5 BEL.1.1.1.5_1   BEL Belgium BEL.1_1 Bruxelles BEL.1.1_1 Bruxelles BEL.1.1.1_1
6 BEL.1.1.1.6_1   BEL Belgium BEL.1_1 Bruxelles BEL.1.1_1 Bruxelles BEL.1.1.1_1
   NAME_3     NAME_4 VARNAME_4  TYPE_4 ENGTYPE_4 CC_4
1 Brussel Anderlecht        NA Commune   Commune   NA
2 Brussel    Brussel        NA Commune   Commune   NA
3 Brussel     Elsene        NA Commune   Commune   NA
```

Belgium Insurance

```
4 Brussel  Etterbeek        NA Commune   Commune   NA
5 Brussel      Evere        NA Commune   Commune   NA
6 Brussel  Ganshoren        NA Commune   Commune   NA
                        geometry           fit
1 MULTIPOLYGON (((4.335771 50...  0.0010460312
2 MULTIPOLYGON (((4.338565 50...            NA
3 MULTIPOLYGON (((4.356911 50... -0.0031532378
4 MULTIPOLYGON (((4.378503 50...  0.0021706600
5 MULTIPOLYGON (((4.390196 50... -0.0016765858
6 MULTIPOLYGON (((4.304328 50...  0.0006273887
```

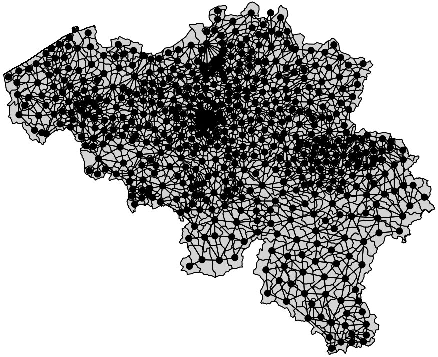
```
R> print(class(Belgium))
```

```
[1] "sf"         "data.frame"
```

Belgium Insurance

Neighborhoodstructures:

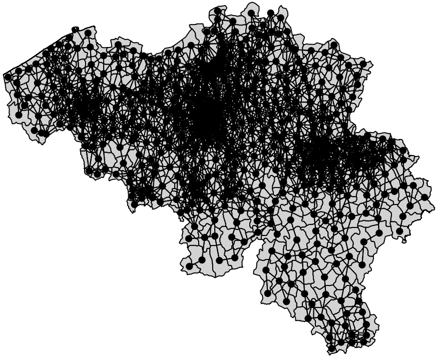
```
R> par(mar = c(0, 0, 0, 0))
R> bamlss::plotneighbors(sf::as_Spatial(Belgium), type = "boundary")
```



Belgium Insurance

Neighborhoodstructures:

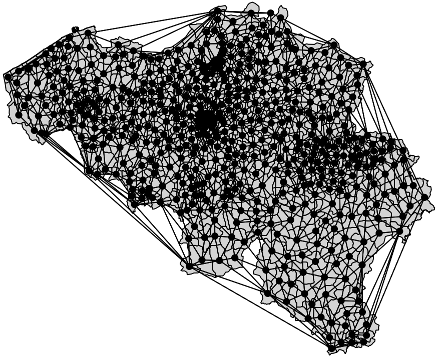
```
R> par(mar = c(0, 0, 0, 0))
R> bamlss::plotneighbors(sf::as_Spatial(Belgium), type = "dist", d1 = 0, d2 = 0.15)
```



Belgium Insurance

Neighborhoodstructures:

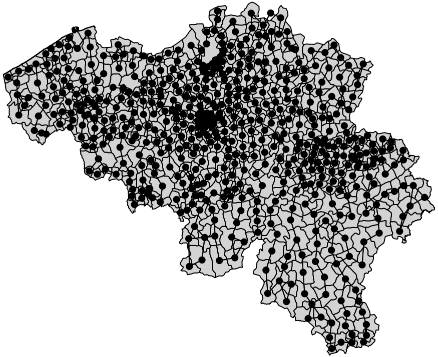
```
R> par(mar = c(0, 0, 0, 0))
R> bamlss::plotneighbors(sf::as_Spatial(Belgium), type = "delaunay")
```



Belgium Insurance

Neighborhoodstructures:

```
R> par(mar = c(0, 0, 0, 0))
R> bamlss::plotneighbors(sf::as_Spatial(Belgium), type = "knear", k = 2)
```



Belgium Insurance

Neighbormatrix:

```
R> nb <- bamlss::neighbormatrix(as_Spatial(Belgium), type = "boundary")
R> print(nb[1:12, 1:20])
    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
1   7 -1  0  0  0  0  0  0  0  0  0 -1 -1  0  0  0  0 -1  0  0
2  -1 16 -1 -1 -1  0 -1  0  0 -1  0 -1 -1 -1  0  0 -1  0 -1  0
3   0 -1  7 -1  0  0  0  0 -1  0  0 -1  0  0  0  0 -1 -1 -1  0
4   0 -1 -1  6  0  0  0  0 -1 -1  0  0  0  0 -1 -1  0  0  0  0
5   0 -1  0  0  4  0  0  0  0 -1  0  0  0  0 -1  0  0  0  0  0
6   0  0  0  0  0  4 -1 -1  0  0 -1  0  0  0  0  0  0  0  0  0
7   0 -1  0  0  0 -1  6 -1  0  0  0  0 -1  0  0  0  0  0  0  0
8   0  0  0  0  0 -1 -1  4  0  0 -1  0 -1  0  0  0  0  0  0  0
9   0  0 -1 -1  0  0  0  0  6  0  0  0  0  0  0 -1  0  0 -1  0
10  0 -1  0 -1 -1  0  0  0  0  5  0  0  0 -1 -1  0  0  0  0  0
11  0  0  0  0  0 -1  0 -1  0  0  5  0 -1  0  0  0  0  0  0  0
12 -1 -1 -1  0  0  0  0  0  0  0  0  4  0  0  0  0  0 -1  0  0
```

Belgium Insurance

Neighbormatrix:

```
R> nb <- bamlss::neighbormatrix(as_Spatial(Belgium), type = "dist", d1 = 0, d2 = 0.1)
R> print(nb[1:12, 1:20])
    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
1  18 -1 -1  0  0 -1 -1 -1  0 -1 -1 -1 -1 -1  0  0 -1 -1  0  0
2  -1 23 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0
3  -1 -1 23 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0
4   0 -1 -1 21 -1 -1 -1 -1 -1 -1  0 -1 -1 -1 -1 -1 -1 -1 -1  0
5   0 -1 -1 -1 21 -1 -1 -1 -1 -1  0 -1 -1 -1 -1 -1 -1  0 -1  0
6  -1 -1 -1 -1 -1 20 -1 -1  0 -1 -1 -1 -1 -1  0  0 -1 -1  0  0
7  -1 -1 -1 -1 -1 -1 19 -1  0 -1 -1 -1 -1 -1  0  0 -1 -1  0  0
8  -1 -1 -1 -1 -1 -1 -1 19  0 -1 -1 -1 -1 -1  0  0 -1 -1  0  0
9   0 -1 -1 -1 -1  0  0  0 19 -1  0 -1  0 -1 -1 -1 -1  0 -1  0
10 -1 -1 -1 -1 -1 -1 -1 -1 -1 23 -1 -1 -1 -1 -1 -1 -1 -1 -1  0
11 -1 -1 -1  0  0 -1 -1 -1  0 -1 17 -1 -1 -1  0  0 -1 -1  0  0
12 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 23 -1 -1 -1 -1 -1 -1 -1  0
```

Belgium Insurance

Neighbormatrix:

```
R> nb <- bamlss::neighbormatrix(as_Spatial(Belgium), type = "knear", k = 2)
R> print(nb[1:12, 1:20])
    1  2  3  4  5  6  7  8 9 10 11 12 13 14 15 16 17 18 19 20
1   3  0  0  0  0  0  0  0 0  0 -1  0  0  0  0  0  0 -1  0  0
2   0  2  0  0  0  0  0  0 0 -1  0  0  0 -1  0  0  0  0  0  0
3   0  0  3 -1  0  0  0  0 0  0  0  0  0 -1  0  0 -1  0  0  0
4   0  0 -1  2  0  0  0  0 0 -1  0  0  0  0  0  0  0  0  0  0
5   0  0  0  0  2  0  0  0 0 -1  0  0  0  0 -1  0  0  0  0  0
6   0  0  0  0  0  5 -1 -1 0  0 -1  0 -1  0  0  0  0  0  0  0
7   0  0  0  0  0 -1  3 -1 0  0  0  0  0  0  0  0  0  0  0  0
8   0  0  0  0  0 -1 -1  3 0  0  0  0 -1  0  0  0  0  0  0  0
9   0  0  0  0  0  0  0  0 2  0  0  0  0  0  0 -1  0  0 -1  0
10  0 -1  0 -1 -1  0  0  0 0  4  0  0  0 -1  0  0  0  0  0  0
11 -1  0  0  0  0 -1  0  0 0  0  4  0 -1  0  0  0  0  0  0  0
12  0  0  0  0  0  0  0  0 0  0  0  2 -1  0  0  0  0 -1  0  0
```

Belgium Insurance

Estimate spatial model. First, we need to make sure that row and column names of the neighboramtrix are the same as the region identifiers in the data.

```
R> nb <- bamlss::neighbormatrix(as_Spatial(Belgium), type = "knear", k = 2)
R> rownames(nb) <- colnames(nb) <- Belgium$NAME_4
```

Now, check if dimensions are correct.

```
R> dim(nb)

[1] 581 581

R> nrow(Belgium)

[1] 581

R> nlevels(BelgiumInsurance$NAME_4)

[1] 554
```

Belgium Insurance

Remove regions not in data.

```
R> i <- which(!(rownames(nb) %in% levels(BelgiumInsurance$NAME_4)))
R> nb2 <- nb[-i, -i]
```

Extra list for smooth constructor `s()`.

```
R> xt <- list("penalty" = nb2)
```

Estimate model.

```
R> if(!file.exists("model.rds")) {
+    b <- bamlss(log(amount) ~ s(NAME_4, bs = "mrf", xt = xt, k = 30),
+      data = BelgiumInsurance)
+    saveRDS(b, file = "model.rds")
+  } else {
+    b <- readRDS("model.rds")
+  }
```

Belgium Insurance

Model summary.

```
R> summary(b)
```

```
Call:
bamlss(formula = log(amount) ~ s(NAME_4, bs = "mrf", xt = xt,
    k = 30), data = BelgiumInsurance)
---
Family: gaussian
Link function: mu = identity, sigma = log
*---
Formula mu:
---
log(amount) ~ s(NAME_4, bs = "mrf", xt = xt, k = 30)
-
Parametric coefficients:
             Mean  2.5%   50% 97.5% parameters
(Intercept) 9.884 9.863 9.883 9.907      9.883
-
```

Belgium Insurance

```
Acceptance probability:
      Mean 2.5% 50% 97.5%
alpha    1    1   1     1
-
Smooth terms:
                    Mean     2.5%      50%    97.5% parameters
s(NAME_4).tau21  0.04882  0.01283  0.04352  0.11494      0.038
s(NAME_4).alpha  1.00000  1.00000  1.00000  1.00000         NA
s(NAME_4).edf   12.91530  6.16710 13.06034 19.23685     12.107
---
Formula sigma:
---
sigma ~ 1
-
Parametric coefficients:
              Mean   2.5%    50%  97.5% parameters
(Intercept) 0.3973 0.3870 0.3973 0.4075      0.397
-
```

Belgium Insurance

```
Acceptance probability:
        Mean   2.5%    50% 97.5%
alpha 0.9972 0.9777 1.0000     1
---
Sampler summary:
-
DIC = 66134.07 logLik = -33059.16 pd = 15.7393
runtime = 29.015
---
Optimizer summary:
-
AICc = 66132.47 edf = 14.1065 logLik = -33052.11
logPost = -33032.93 nobs = 18203 runtime = 0.211
```

Belgium Insurance

Predict spatial effect.

```
R> nd <- unique(BelgiumInsurance[, "NAME_4", drop = FALSE])
R> nd$fit <- predict(b, newdata = nd,
+    model = "mu", term = "s(NAME_4)",
+    intercept = FALSE, FUN = mean)
R> print(head(nd))
```

```
                  NAME_4          fit
1   Sint-Joost-ten-Node -0.009074526
198          Schaarbeek -0.008223255
357           Etterbeek -0.013451474
445              Elsene -0.017886475
558         Sint-Gillis -0.081380162
627          Anderlecht -0.074994321
```

```
R> print(dim(nd))
```

```
[1] 554   2
```

Belgium Insurance

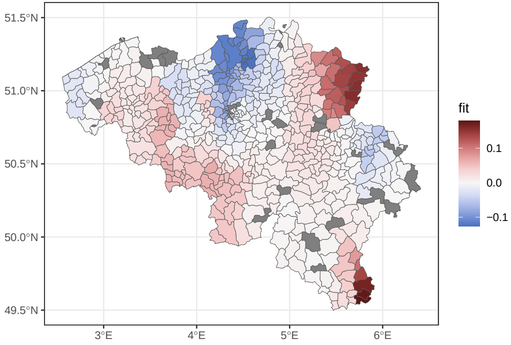
We need to match the predictions to the regions in the map.

```
R> Belgium$fit <- NA
R> for(j in seq_along(nd$NAME_4))
+    Belgium$fit[Belgium$NAME_4 == nd$NAME_4[j]] <- nd$fit[j]
```

Belgium Insurance

Plot effect.

```
R> library("ggplot2")
R> ggplot(Belgium) + geom_sf(aes(fill = fit)) +
+  scale_fill_continuous_diverging("Blue-Red 3") + theme_bw()
```



Belgium Insurance

Estimates for missing regions.

```
R> ## NA regions.
R> j <- Belgium$NAME_4[is.na(Belgium$fit)]

R> for(i in seq_along(j)) {
+     ## Neighbors of NA region.
+     n <- nb[j[i], ]
+     n <- which(n < 0)
+
+     ## Get Index.
+     l <- Belgium$NAME_4 == j[i]
+
+     ## Compute mean of neighbors.
+     Belgium$fit[l] <- mean(Belgium$fit[Belgium$NAME_4 %in% names(n)], na.rm = TRUE)
+   }
```

Belgium Insurance

Full spatial effect.

```
R> ggplot(Belgium) + geom_sf(aes(fill = fit)) +
+  scale_fill_continuous_diverging("Blue-Red 3") + theme_bw()
```