Example 3.4

Disease mapping: from foundations to multidimensional modeling

Martinez-Beneito M.A. and Botella-Rocamora P.

This document reproduces the analysis made at Example 3.4 of the book: "Disease mapping: from foundations to multidimensional modeling" by Martinez-Beneito M.A. and Botella-Rocamora P., published by CRC press in 2019. You can watch the analysis made with full detail at this pdf document, or even execute it if you want with the material available at https://github.com/MigueBeneito/DMBook. Anyway, this pdf file should be enough for following most of the details of the analysis made for this example.

The statistical analysis below has been run in `R`, by additionally using the library `Rmarkdown`, so be sure that you have this software installed if you want to reproduce by yourself the content of this document. In that case we advise you to download first the annex material at https://github.com/MigueBeneito/DMBook, open with `Rstudio` the corresponding `.Rproj` file that you will find at the folder corresponding to this example and compile the corresponding `.Rmd` document. This will allow you to reproduce the whole statistical analysis below.

Libraries and data loading

```
# Libraries loading
#-----------------
if (!require(R2WinBUGS)) {
    install.packages("R2WinBUGS")
    library(R2WinBUGS)
}
if (!require(INLA)) {
    install.packages("INLA", repos = c(getOption("repos"), INLA = "https://inla.r-inla-download.org/R/s
        dep = TRUE)
    library(INLA)
}
if (!require(pbugs)) {
    if (!require(devtools)) {
        install.packages("devtools")
        devtools::install_github("fisabio/pbugs")
    } else {
        install_github("fisabio/pbugs")
    }
}

# Data loading
#------------
load("../Data/OralCancerTimeTrends.RData")
```

Data preparation

```
# covariate
year = 1991:2011
year.centered = year - mean(year)
```

```r
# rates
rate = 1e+05 * O/Pop
```

INLA call to splines model with default priors

```r
# basis of functions of 9 elements
base9 = matrix(nrow = 21, ncol = 9)
for (j in 1:9) {
    base9[, j] = dnorm(1:21, 1 + j * 2, 2)
}

# INLA call
data = data.frame(rate = rate, year = year - 2001, id.node = 1:21)
form1 = rate ~ year + f(id.node, model = "z", Z = base9)
result1 = inla(form1, data = data, control.compute = list(dic = TRUE))

# results summary
summary(result1)
```

```
##
## Call:
## "inla(formula = form1, data = data, control.compute = list(dic = TRUE))"
##
## Time used:
##  Pre-processing    Running inla Post-processing           Total
##          1.7207          1.1688          0.0932          2.9827
##
## Fixed effects:
##                mean     sd 0.025quant 0.5quant 0.975quant    mode kld
## (Intercept)  7.7585 0.1462     7.4687   7.7585     8.0480  7.7585   0
## year        -0.1236 0.0241    -0.1714  -0.1236    -0.0758 -0.1236   0
##
## Random effects:
## Name   Model
##  id.node   Z model
##
## Model hyperparameters:
##                                              mean        sd 0.025quant
## Precision for the Gaussian observations     2.457 7.587e-01      1.244
## Precision for id.node                   18604.889 1.836e+04   1262.188
##                                          0.5quant 0.975quant     mode
## Precision for the Gaussian observations     2.369      4.202    2.197
## Precision for id.node                   13185.481  67158.639 3448.492
##
## Expected number of effective parameters(std dev): 2.001(0.0032)
## Number of equivalent replicates : 10.49
##
## Deviance Information Criterion (DIC) ...............: 46.79
## Deviance Information Criterion (DIC, saturated) ....: 27.15
## Effective number of parameters .....................: 3.076
##
## Marginal log-Likelihood:  -37.94
```

```
## Posterior marginals for linear predictor and fitted values computed
```

WinBUGS call to random effects splines model

```
# WinBUGS model
model.random = function() {
    for (i in 1:n) {
        rate[i] ~ dnorm(mu[i], tau.rate)
        mu[i] <- beta[1] + beta[2] * year[i] + inprod2(base[i, ], gamma[])
    }
    for (i in 1:nBase) {
        gamma[i] ~ dnorm(0, tau.random)
    }
    beta[1] ~ dflat()
    beta[2] ~ dflat()

    tau.rate <- pow(sd.rate, -2)
    tau.random <- pow(sd.random, -2)
    sd.rate ~ dunif(0, 10)
    sd.random ~ dunif(0, 10)
}

# WinBUGS run with a basis of 9 functions
dataWB = list(rate = rate, year = year.centered, base = base9, nBase = 9,
    n = 21)
inits = function() {
    list(beta = rnorm(2), gamma = rnorm(9))
}
param = c("beta", "gamma", "sd.rate", "sd.random", "mu")

time.bugs = system.time(ResulWB <- bugs(data = dataWB, inits = inits, param = param,
    model = model.random, n.iter = 10000, n.burnin = 1000, n.thin = 9,
    DIC = FALSE, bugs.seed = 1))

# We call WinBUGS twice, with bugs and pbugs functions, for comparing
# the computational performance of inla and WinBUGS
time.pbugs = system.time(ResulWB <- pbugs(data = dataWB, inits = inits,
    param = param, model = model.random, n.iter = 10000, n.burnin = 1000,
    n.thin = 9, DIC = FALSE, bugs.seed = 1))
```

Alternative INLA models

```
# Gamma(2,0.00005) prior on the precision parameter of the splines
# coefficients
form1.b = rate ~ year + f(id.node, model = "z", Z = base9, hyper = list(prec = list(param = c(2,
    5e-05))))
result1.b = inla(form1.b, data = data, control.compute = list(dic = TRUE))
summary(result1.b)
```

```
##
## Call:
```

```
## "inla(formula = form1.b, data = data, control.compute = list(dic = TRUE))"
##
## Time used:
##  Pre-processing    Running inla Post-processing           Total
##          1.3816          0.9509          0.0780          2.4105
##
## Fixed effects:
##                mean     sd 0.025quant 0.5quant 0.975quant    mode kld
## (Intercept)  7.7585 0.1464     7.4684   7.7585     8.0483  7.7585   0
## year        -0.1236 0.0242    -0.1715  -0.1236    -0.0757 -0.1236   0
##
## Random effects:
## Name   Model
##  id.node   Z model
##
## Model hyperparameters:
##                                              mean        sd 0.025quant
## Precision for the Gaussian observations     2.457 7.587e-01      1.244
## Precision for id.node                   38929.455 2.748e+04   6696.922
##                                          0.5quant 0.975quant      mode
## Precision for the Gaussian observations     2.369  4.202e+00     2.197
## Precision for id.node                   32463.028  1.090e+05 18722.264
##
## Expected number of effective parameters(std dev): 2.00(2e-04)
## Number of equivalent replicates : 10.50
##
## Deviance Information Criterion (DIC) ...............: 46.81
## Deviance Information Criterion (DIC, saturated) ....: 27.17
## Effective number of parameters .....................: 3.083
##
## Marginal log-Likelihood:  -37.91
## Posterior marginals for linear predictor and fitted values computed
```

```r
# Uniform prior on the standard deviation parameter of the splines
# coefficients
sdunif = "expression:
    logdens=-log_precision/2;
    return(logdens)"
formula2 = rate ~ year + f(id.node, model = "z", Z = base9, hyper = list(prec = list(prior = sdunif)))
result2 <- inla(formula2, data = data, control.compute = list(dic = TRUE),
    control.family = list(hyper = list(prec = list(prior = sdunif))))
summary(result2)
```

```
##
## Call:
## c("inla(formula = formula2, data = data, control.compute = list(dic = TRUE), ",  "    control.family
##
## Time used:
##  Pre-processing    Running inla Post-processing           Total
##          1.1920          0.9689          0.0860          2.2469
##
## Fixed effects:
##                mean     sd 0.025quant 0.5quant 0.975quant    mode   kld
## (Intercept)  7.9096 0.3834     7.3155   7.8482     8.8796  7.7892 1e-04
## year        -0.1129 0.0424    -0.1926  -0.1151    -0.0207 -0.1183 0e+00
```

```
##
## Random effects:
## Name   Model
##  id.node   Z model
##
## Model hyperparameters:
##                                          mean      sd 0.025quant 0.5quant
## Precision for the Gaussian observations 2.258  0.8183     1.0116   2.1447
## Precision for id.node                   3.155 26.8913     0.0198   0.4382
##                                         0.975quant   mode
## Precision for the Gaussian observations       4.19 1.9192
## Precision for id.node                        21.14 0.0385
##
## Expected number of effective parameters(std dev): 3.895(1.514)
## Number of equivalent replicates : 5.392
##
## Deviance Information Criterion (DIC) ...............: 48.36
## Deviance Information Criterion (DIC, saturated) ....: 26.75
## Effective number of parameters .....................: 5.454
##
## Marginal log-Likelihood:  -27.49
## Posterior marginals for linear predictor and fitted values computed
```

```
# RW1 prior (with uniform prior on the standard deviation) for the year
# effect
formula3 = rate ~ f(id.node, model = "rw1", hyper = list(theta = list(prior = sdunif)))
result3 <- inla(formula3, data = data, control.compute = list(dic = TRUE),
    control.family = list(hyper = list(theta = list(prior = sdunif))))
summary(result3)
```

```
##
## Call:
## c("inla(formula = formula3, data = data, control.compute = list(dic = TRUE), ",  "    control.family
##
## Time used:
##  Pre-processing    Running inla Post-processing           Total
##          1.0669          0.8120          0.0750          1.9539
##
## Fixed effects:
##               mean     sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) 7.7585 0.1401      7.476   7.7585      8.041 7.7585   0
##
## Random effects:
## Name   Model
##  id.node   RW1 model
##
## Model hyperparameters:
##                                          mean    sd 0.025quant 0.5quant
## Precision for the Gaussian observations 2.752 1.254      1.023    2.521
## Precision for id.node                   9.204 9.305      1.264    6.469
##                                         0.975quant  mode
## Precision for the Gaussian observations      5.836 2.093
## Precision for id.node                       33.794 3.243
##
## Expected number of effective parameters(std dev): 8.034(3.734)
```

```
## Number of equivalent replicates : 2.614
##
## Deviance Information Criterion (DIC) ...............: 44.69
## Deviance Information Criterion (DIC, saturated) ....: 26.26
## Effective number of parameters .....................: 7.612
##
## Marginal log-Likelihood:  -26.33
## Posterior marginals for linear predictor and fitted values computed
```

```r
# RW2 prior (with uniform prior on the standard deviation) for the year
# effect
formula4 = rate ~ f(id.node, model = "rw2", hyper = list(theta = list(prior = sdunif)))
result4 <- inla(formula4, data = data, control.compute = list(dic = TRUE),
    control.family = list(hyper = list(theta = list(prior = sdunif))))
summary(result4)
```

```
##
## Call:
## c("inla(formula = formula4, data = data, control.compute = list(dic = TRUE), ",  "    control.family
##
## Time used:
##  Pre-processing    Running inla Post-processing           Total
##          1.0698          0.6931          0.0740          1.8368
##
## Fixed effects:
##               mean     sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) 7.7585 0.1568     7.4465   7.7585     8.0701 7.7585   0
##
## Random effects:
## Name   Model
##  id.node   RW2 model
##
## Model hyperparameters:
##                                             mean        sd 0.025quant
## Precision for the Gaussian observations    2.072 7.256e-01     0.9188
## Precision for id.node                   8439.108 3.486e+06     8.2737
##                                         0.5quant 0.975quant   mode
## Precision for the Gaussian observations     1.99      3.728  1.812
## Precision for id.node                     244.31  39609.610 15.319
##
## Expected number of effective parameters(std dev): 3.859(1.695)
## Number of equivalent replicates : 5.442
##
## Deviance Information Criterion (DIC) ...............: 48.65
## Deviance Information Criterion (DIC, saturated) ....: 26.21
## Effective number of parameters .....................: 5.287
##
## Marginal log-Likelihood:  -27.57
## Posterior marginals for linear predictor and fitted values computed
```

Computational comparison between INLA and WinBUGS

```r
result2$cpu.used
```

```
##  Pre-processing    Running inla Post-processing           Total
##      1.19203615      0.96888089      0.08595109      2.24686813
```

```r
time.bugs
```

```
##    user  system elapsed
##    0.22    0.03    8.34
```

```r
time.pbugs
```

```
##    user  system elapsed
##    0.29    0.09    5.97
```

DIC comparison between several INLA models

```r
# splines model
result2$dic$dic
```

```
## [1] 48.36328
```

```r
# RW1
result3$dic$dic
```

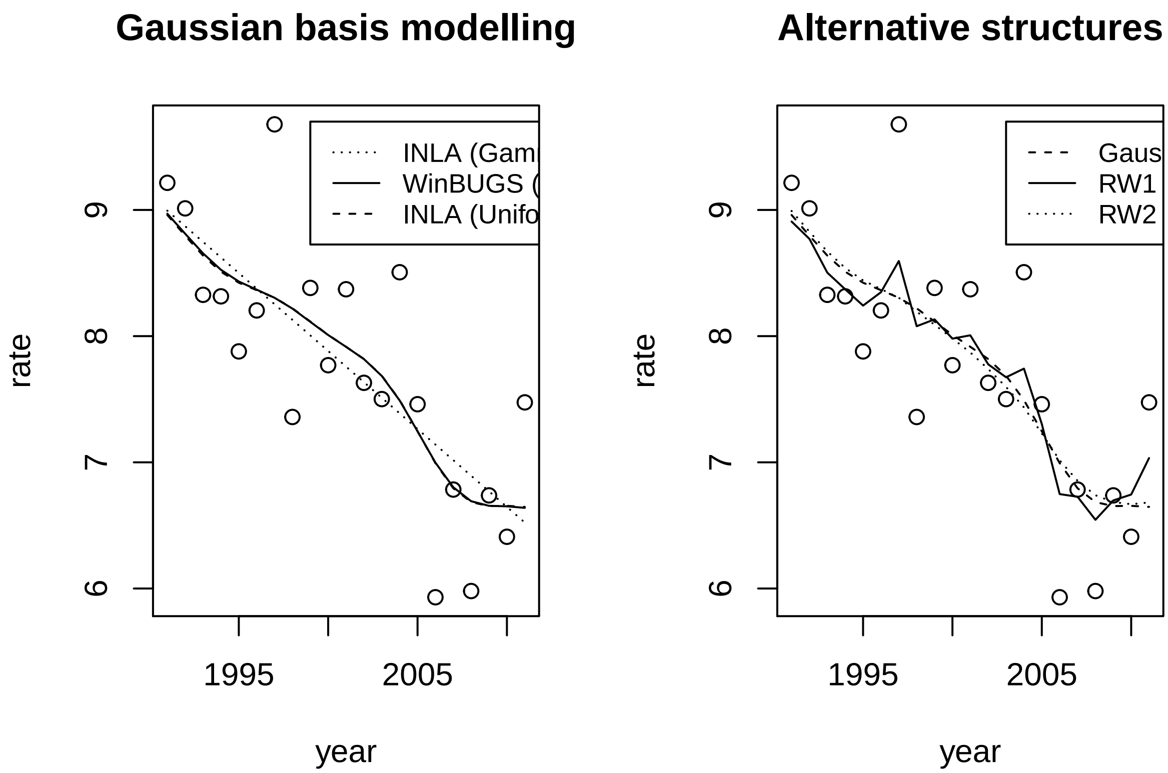
```
## [1] 44.69484
```

```r
# RW2
result4$dic$dic
```

```
## [1] 48.64895
```

Plot illustrating the different fits made

```r
par(mfrow = c(1, 2))
plot(year, rate)
lines(year, ResulWB$mean$mu)
lines(year, result2$summary.fitted.values[, 1], lty = 2)
lines(year, result1$summary.fitted.values[, 1], lty = 3)
legend(1999, 9.7, lty = c(3, 1, 2), legend = c("INLA (Gamma prior)", "WinBUGS (Uniform prior)",
    "INLA (Uniform prior)"), cex = 0.8)
title("Gaussian basis modelling")
plot(year, rate)
lines(year, result2$summary.fitted.values[, 1], lty = 2)
lines(year, result3$summary.fitted.values[, 1], lty = 1)
lines(year, result4$summary.fitted.values[, 1], lty = 3)
legend(2003, 9.7, lty = c(2, 1, 3), legend = c("Gaussian basis", "RW1",
    "RW2"), cex = 0.8)
title("Alternative structures")
```


Gaussian basis modelling
INLA (Gam
WinBUGS (
INLA (Unifo
rate
9
8
7
6
1995
2005
year
Alternative structures
Gaus
RW1
RW2
rate
9
8
7
6
1995
2005
year