Package 'INLAMSM'

June 4, 2021

Type Package

Title Multivariate Spatial Models with 'INLA'

Version 0.2-3

Maintainer Virgilio Gómez-Rubio <Virgilio.Gomez@uclm.es>

Description Implementation of several multivariate areal latent effects for 'INLA' using the 'rgeneric' latent effect (Palmí-Perales et al., 2019, <doi:10.18637/jss.v098.i02>). The 'INLA' package can be downloaded from <https://www.r-inla.org>. In particular, the package includes latent effects ready to use for several multivariate spatial models: intrinsic CAR, proper CAR and the M-model (Botella-Rocamora et al., 2015, <doi:10.1002/sim.6423>).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 3.5.0), Matrix, MCMCpack, sp, spdep

Suggests INLA, spData, rgdal

Additional_repositories https://inla.r-inla-download.org/R/stable

NeedsCompilation no

Author Francisco Palmí-Perales [aut],
Virgilio Gómez-Rubio [aut, cre],
Miguel A. Martinez-Beneito [aut]

Repository CRAN

Date/Publication 2021-06-04 08:40:06 UTC

R topics documented:

`CV` *Multivariate mortality data from Comunidad Valenciana (Spain)*

Description

Simulated multivariate mortality data from Comunidad Valenciana (Spain). The data set contains (simulated) observed and expected deaths for Cirrhosis, Lung cancer and Cirrhosis for the Valencian municipalities. The supplied data have been simulated mimicking the original data set which has privacy restrictions. Additional details on the generation of the supplied dataset can be found at the original book.

Usage

```
data(CV)
```

Format

A `SpatialPolygonsDataFrame` with the boundaries of the municipalities in Comunidad Valenciana with the following columns:

CODMUNI Municipality code.

NOMBRE Name of the municipality.

Exp.Cirrhosis Expected number of cases of cirrhosis.

Exp.Lung Expected number of cases of lung cancer.

Exp.Oral Expected number of cases of oral cavity cancer.

Obs.Cirrhosis Observed number of cases of cirrhosis.

Obs.Lung Observed number of cases of lung cancer.

Obs.Oral Observed number of cases of oral cavity cancer.

Source

The original data set is supplied as supplementary material of the book: "Martinez-Beneito, M A & Botella Rocamora, P. Disease mapping: from foundations to multidimensional modeling. CRC/Chapman & Hall, 2019". This object has been built from several of the files available at the supplementary material repository of the book at: https://github.com/MigueBeneito/DisMapBook/tree/master/Data

References

Martinez-Beneito, M A & Botella Rocamora, P. Disease mapping: from foundations to multidimensional modeling. CRC/Chapman & Hall, 2019.

Palmí-Perales F, Gómez-Rubio V, Martinez-Beneito MA (2021). "Bayesian Multivariate Spatial Models for Lattice Data with INLA." _Journal of Statistical Software_, *98*(2), 1-29. doi: 10.18637/jss.v098.i02 (URL: https://doi.org/10.18637/jss.v098.i02).

See Also

CV.nb

Examples

```
if(require(INLA, quietly = TRUE)) {
require(sp)
require(spdep)
data(CV)
W <- as(nb2mat(CV.nb, style = "B"), "Matrix")

#Data (two diseases only)
d <- list(OBS = c(CV$Obs.Cirrhosis, CV$Obs.Lung),
 EXP = c(CV$Exp.Cirrhosis, CV$Exp.Lung))

 # Index for latent effect
d$idx <- 1:length(d$OBS)

k <- 2  #Number of diseases

# Linear constraint for models
A <- kronecker(Diagonal(k, 1), Matrix(1, ncol = nrow(W), nrow = 1))
e = rep(0, k)

# Two independent ICAR models
#model <- inla.rgeneric.define(inla.rgeneric.indep.IMCAR.model,
#  k = k, W = W)
model <- inla.INDIMCAR.model(k = k, W = W)
r.simcar <- try(
  inla(OBS ~ 1 + f(idx, model = model, extraconstr = list(A = as.matrix(A), e = e)),
    data = d, E = EXP, family = "poisson",
     # To run faster, REMOVE in real applications
     control.mode = list(theta = c(1.4, 2.1), restart = TRUE),
    control.predictor = list(compute = TRUE))
)
summary(r.simcar)

# IMCAR model
#model <- inla.rgeneric.define(inla.rgeneric.IMCAR.model,
#  k = k, W = W, alpha.min = 0, alpha.max = 1)
model <- inla.IMCAR.model(k = k, W = W)
```

```
r.imcar <- try(
  inla(OBS ~ 1 + f(idx, model = model, extraconstr = list(A = as.matrix(A), e = e)),
    data = d, E = EXP, family = "poisson",
     # To run faster, REMOVE in real applications
     control.mode = list(theta = c(1.77, 2.01, 0.93),
       restart = TRUE),
    control.compute = list(config = TRUE),
    control.predictor = list(compute = TRUE))
)
summary(r.imcar)

# Transform parameters
summary.post <- inla.MCAR.transform(r.imcar, k = k)

# Posterior of variance matrix
summary.post$VAR.p # Using point estimates
summary.post$VAR.m # Using posterior sampling

} #if(require(INLA))
```

CV.nb *Adjacency matrix of municipalities in Comunidad Valenciana (Spain)*

Description

Adjacency matrix of municipalities in Comunidad Valenciana (Spain)

Usage

```
data(CV)
```

Format

An nb object with the adjacencies of the municipalities in Comunidad Valenciana (Spain) to be used for the spatial models.

Source

The original data set is supplied as supplementary material of the book: "Martinez-Beneito, M A & Botella Rocamora, P. Disease mapping: from foundations to multidimensional modeling. CRC/Chapman & Hall, 2019". This object has been built from several of the files available at the supplementary material repository of the book at: https://github.com/MigueBeneito/DisMapBook/tree/master/Data

References

Martinez-Beneito, M A & Botella Rocamora, P. Disease mapping: from foundations to multidimensional modeling. CRC/Chapman & Hall, 2019.

See Also

CV

Examples

```
require(sp)
require(spdep)
data(CV)
plot(CV)
plot(CV.nb, coordinates(CV), pch = ".", col = "gray", add = TRUE)
```

`inla.MCAR.transform` *Transform hyperparameters in multivarite spatial models.*

Description

Multivariate spatial models fit will report hyperparameters in the internal scale. These functions will transform the hyperparameters to a different scale. Using this function requires setting `control.compute = list(config = TRUE)` when fitting the model with INLA.

Usage

```
inla.MCAR.transform(obj, k, model = "IMCAR", alpha.min, alpha.max)
```

Arguments

`obj`	An 'inla' object with an MCAR, IMCAR or M-model latent effect.
`k`	Number of variables in the multivariate model.
`model`	Either "INDIMCAR", "INDPMCAR", "IMCAR" or "PMCAR". Not used for M-models.
`alpha.min`	Lower bound of the autocorrelation parameter alpha.
`alpha.max`	Upper bound of the autocorrelation parameter alpha.

Value

This function returns a list with the following elements:

- `marginals.hyperpar` List with the posterior marginals of transformed hyperparameters.
- *summary.hyperpar* Summary of the posterior marginals.
- *VAR.p* Variance matrix of between-variables variability computed using point estimates from the posterior marginals.
- *VAR.m* Posterior mean of variance matrix of the between-variables variability. This is computed using the internal representation of the posterior joint distribution of the hyperparameters.

- *confs* Configurations of the hyperparameters used to compute *VAR.m*. This is obtained from `obj$misc$configs$config`.
- *M.p* M matrix (only in the M-model) obtained using point estimates of the parameters.
- *M.m* M matrix (only in the M-model) obtained using the the internal representation of the posterior joint distribution of the hyperparameters.

References

Palmí-Perales F, Gómez-Rubio V, Martinez-Beneito MA (2021). "Bayesian Multivariate Spatial Models for Lattice Data with INLA." _Journal of Statistical Software_, *98*(2), 1-29. doi: 10.18637/jss.v098.i02 (URL: https://doi.org/10.18637/jss.v098.i02).

`inla.rgeneric.IMCAR.model`

$IMCAR(\Lambda)$: Intrinsic multivariate CAR latent effect.

Description

Multivariate generalization of the intrinsic conditional autorregresive model. The matrix which models the variability between diseases is a symmetric matrix with the inverse of the marginal precisions on the diagonal elements and the correlation parameters divided by the square root of the precisions on the off-diagonal elements.

Usage

```
inla.rgeneric.IMCAR.model(cmd, theta)

inla.IMCAR.model(...)
```

Arguments

`...`	Arguments to be passed to 'inla.rgeneric.define'.
`cmd`	Arguments used by latent effects defined using the 'rgeneric' latent effect.
`theta`	Vector of hyperparameters.

Details

This function is used to define a latent effect that is a multivariate spatial effect with a intrinsic conditional autorregresive distribution and a symmetric matrix in order to model the whitin-disease and the between-diseases variability, respectively. Due to this effect is a multivariate spatial latent effect this function requires the following arguments when defining the latent effect:

- *W* Adjacency SPARSE matrix for spatial effect in the basic binary code.
- *k* Number of diseases of the multivariate study.

This model is defined using the 'f()' function and an index in order to identify the spatial areas. See the example.

Value

This is used internally by the 'INLA::inla()'.

Prior distributions of the hyperparameters

The hyperparamenters of this lattent effect are the marginal precisions of each disease which are equal to the number of diseases and the correlation parameters for the whole pair of diseases.

References

Palmí-Perales F, Gómez-Rubio V, Martinez-Beneito MA (2021). "Bayesian Multivariate Spatial Models for Lattice Data with INLA." _Journal of Statistical Software_, *98*(2), 1-29. doi: 10.18637/jss.v098.i02 (URL: https://doi.org/10.18637/jss.v098.i02).

Examples

```
if (require("INLA", quietly = TRUE)) {
require(spdep)
require(spData)
require(rgdal)

#Load SIDS data
nc.sids <- readOGR(system.file("shapes/sids.shp", package="spData")[1])
proj4string(nc.sids) <- CRS("+proj=longlat +ellps=clrk66")

#Compute adjacency matrix, as nb object 'adj' and sparse matrix 'W'
adj <- poly2nb(nc.sids)
W <- as(nb2mat(adj, style = "B"), "Matrix")

#Compute expected cases
r74 <- sum(nc.sids$SID74) / sum(nc.sids$BIR74)
nc.sids$EXP74 <- r74 * nc.sids$BIR74
nc.sids$SMR74 <- nc.sids$SID74 / nc.sids$EXP74
nc.sids$NWPROP74 <- nc.sids$NWBIR74 / nc.sids$BIR74

r79 <- sum(nc.sids$SID79) / sum(nc.sids$BIR79)
nc.sids$EXP79 <- r79 * nc.sids$BIR79
nc.sids$SMR79 <- nc.sids$SID79 / nc.sids$EXP79
nc.sids$NWPROP79 <- nc.sids$NWBIR79 / nc.sids$BIR79

# Define sum-to-zero constraints
A <- kronecker(Diagonal(2, 1), Matrix(1, ncol = nrow(W), nrow = 1))
e  = rep(0, 2)

# Data (replicated to assess scalability)

#Real data
n.rep <- 1
d <- list(OBS = c(nc.sids$SID74, nc.sids$SID79),
          NWPROP = c(nc.sids$NWPROP74, nc.sids$NWPROP79),
```

```
          EXP = c(nc.sids$EXP74, nc.sids$EXP79))
d <- lapply(d, function(X) { rep(X, n.rep)})
d$idx <- 1:length(d$OBS)

# Model parameters
k <- 2 * n.rep #Number of diseases


#Define model IMCAR
model <- inla.rgeneric.define(inla.rgeneric.IMCAR.model, debug = FALSE,
  k = k, W = W)

#Fit model
r <- inla(OBS ~ 1 + f(idx, model = model,
    extraconstr = list(A = as.matrix(A), e = e)), # + NWPROP,
  data = d, E = EXP, family = "poisson",
  control.compute = list(config = TRUE),
  control.predictor = list(compute = TRUE))

summary(r)

# Transformed parameters
r.hyperpar <- inla.MCAR.transform(r, k = 2, model = "IMCAR")
r.hyperpar$summary.hyperpar

#Get fitted data, i.e., relative risk
nc.sids$FITTED74 <- r$summary.fitted.values[1:100, "mean"]
nc.sids$FITTED79 <- r$summary.fitted.values[100 + 1:100, "mean"]

#Display fitted relative risks
dev.new()
spplot(nc.sids, c("SMR74", "FITTED74", "SMR79", "FITTED79"))

#Show marginals of tau_1, tau_2, rho

marg.tau1 <- inla.tmarginal(
  function(x) exp(x),
  r$marginals.hyperpar[[1]])

marg.tau2 <- inla.tmarginal(
  function(x) exp(x),
  r$marginals.hyperpar[[2]])

marg.rho <- inla.tmarginal(
  function(x) (2*exp(x))/(1 + exp(x)) - 1,
  r$marginals.hyperpar[[3]])


dev.new()

oldpar <- par(mfrow = c(2, 2))
plot(marg.tau1, main = "tau1", type = "l")
plot(marg.tau2, main = "tau2", type = "l")
```

```
plot(marg.rho, main = "rho", type = "l")

par(oldpar)

## Running UNIVARIATE MODEL

#Real data
n.rep <- 1
d <- list(OBS = nc.sids$SID74,
          NWPROP = nc.sids$NWPROP74,
          EXP = nc.sids$EXP74)
d <- lapply(d, function(X) { rep(X, n.rep)})
d$idx <- 1:length(d$OBS)

#Fit model
r.uni <- inla(OBS ~ 1 + f(idx, model = "besag", graph = W), # + NWPROP,
              data = d, E = EXP, family = "poisson",
              control.predictor = list(compute = TRUE))

summary(r.uni)

nc.sids$FITTED74.uni <- r.uni$summary.fitted.values[ , "mean"]

#Display univariate VS multivariate  fitted relative risks.
dev.new()
spplot(nc.sids, c("SMR74", "FITTED74", "FITTED74.uni"))
spplot(nc.sids, c("FITTED74", "FITTED74.uni"),
       main=list(label="Relative risk estimation",cex=2))
dev.new()
plot(nc.sids$FITTED74.uni, nc.sids$FITTED74,
     main="Relative Risk estimations", xlab="Univariate RR estimations",
     ylab="Multivariate RR estimations", xlim=c(0.5, 2.5), ylim=c(0.5, 2.5))
abline(h=0, col="grey")
abline(v=0, col="grey")
abline(a=0, b=1, col="red")

#Plot posterior mean of the spatial effects univ VS multi

nc.sids$m.uni <- r.uni$summary.random$idx[, "mean"]
nc.sids$m.mult <- r$summary.random$idx[1:100, "mean"]
dev.new()
plot(nc.sids$m.uni, nc.sids$m.mult,
     main="Posterior mean of the spatial effect", xlab="Uni. post. means"
     , ylab="Mult. post. means", xlim=c(-1,1), ylim=c(-1,1))
abline(h=0, col="grey")
abline(v=0, col="grey")
abline(a=0, b=1, col="red")

dev.new()
spplot(nc.sids, c("m.mult", "m.uni"),
       main=list(label="Post. mean spatial effect",cex=2))
}
```

`inla.rgeneric.indep.IMCAR.model` *$MCAR(1, \Lambda)$: Intrinsic multivariate CAR latent effect without correlation parameters.*

Description

Multivariate generalization of the intrinsic conditional autorregresive model. No correlation parameters are considered between the different diseases, so the matrix which models the variability between diseases will be a diagonal matrix.

Usage

```
inla.rgeneric.indep.IMCAR.model(cmd, theta)

inla.INDIMCAR.model(...)
```

Arguments

`...`	Arguments to be passed to 'inla.rgeneric.define'.
`cmd`	Arguments used by latent effects defined using the 'rgeneric' latent effect.
`theta`	Vector of hyperparameters.

Details

This function is used to define a latent effect that is a multivariate spatial effect with a intrinsic conditional autorregresive distribution and a diagonal matrix in order to model the whitin-disease and the between-diseases variability, respectively. Due to this effect is a multivariate spatial latent effect this function requires the following arguments when defining the latent effect:

- *W* Adjacency SPARSE matrix for spatial effect in the basic binary code.
- *k* Number of diseases of the multivariate study.

This model is defined using the 'f()' function and an index in order to identify the spatial areas. See the example.

Value

This is used internally by the 'INLA::inla()'.

Prior distributions of the hyperparameters

The hyperparamenters of this lattent effect are the marginal precisions of each disease. So the total number of hyperpameters is equal to the number of diseases.

References

Palmí-Perales F, Gómez-Rubio V, Martinez-Beneito MA (2021). "Bayesian Multivariate Spatial Models for Lattice Data with INLA." _Journal of Statistical Software_, *98*(2), 1-29. doi: 10.18637/jss.v098.i02 (URL: https://doi.org/10.18637/jss.v098.i02).

Examples

```
if (require("INLA", quietly = TRUE)) {
require(spdep)
require(spData)
require(rgdal)

## Independent IMCAR model with 2 diseases

#Load SIDS data
nc.sids <- readOGR(system.file("shapes/sids.shp", package="spData")[1])
proj4string(nc.sids) <- CRS("+proj=longlat +ellps=clrk66")

#Compute adjacency matrix, as nb object 'adj' and sparse matrix 'W'
adj <- poly2nb(nc.sids)
W <- as(nb2mat(adj, style = "B"), "Matrix")

#Compute expected cases
r74 <- sum(nc.sids$SID74) / sum(nc.sids$BIR74)
nc.sids$EXP74 <- r74 * nc.sids$BIR74
nc.sids$SMR74 <- nc.sids$SID74 / nc.sids$EXP74
nc.sids$NWPROP74 <- nc.sids$NWBIR74 / nc.sids$BIR74

r79 <- sum(nc.sids$SID79) / sum(nc.sids$BIR79)
nc.sids$EXP79 <- r79 * nc.sids$BIR79
nc.sids$SMR79 <- nc.sids$SID79 / nc.sids$EXP79
nc.sids$NWPROP79 <- nc.sids$NWBIR79 / nc.sids$BIR79

# Data (replicated to assess scalability)

#Real data
n.rep <- 1
d <- list(OBS = c(nc.sids$SID74, nc.sids$SID79),
         NWPROP = c(nc.sids$NWPROP74, nc.sids$NWPROP79),
         EXP = c(nc.sids$EXP74, nc.sids$EXP79))
d <- lapply(d, function(X) { rep(X, n.rep)})
d$idx <- 1:length(d$OBS)

# Model parameters: k and W
k <- 2 * n.rep #Number of diseases

#Define independent IMCAR model
model <- inla.rgeneric.define(inla.rgeneric.indep.IMCAR.model, debug = FALSE,
                              k = k,
                              W = W)
```

```
# Matrices for sum-to-zero constraints
A <- kronecker(Diagonal(k, 1), Matrix(1, ncol = nrow(W), nrow = 1))
e  = rep(0, k)

#Fit multivariate model
r <- inla(OBS ~ 1 + f(idx, model = model,
      extraconstr = list(A = as.matrix(A), e = e)), # + NWPROP,
 data = d, E = EXP, family = "poisson",
 control.predictor = list(compute = TRUE))

summary(r)

# Transformed parameters
r.hyperpar <- inla.MCAR.transform(r, k = 2, model = "INDIMCAR")
r.hyperpar$summary.hyperpar

#Get fitted data, i.e., relative risk
nc.sids$FITTED74 <- r$summary.fitted.values[1:100, "mean"]
nc.sids$FITTED79 <- r$summary.fitted.values[100 + 1:100, "mean"]

#Display fitted relative risks
dev.new()
spplot(nc.sids, c("SMR74", "FITTED74", "SMR79", "FITTED79"))

#Show marginals of tau_1, tau_2, alpha

marg.tau1 <- inla.tmarginal(
 function(x) exp(x),
 r$marginals.hyperpar[[1]])

marg.tau2 <- inla.tmarginal(
 function(x) exp(x),
 r$marginals.hyperpar[[2]])

oldpar <- par(mfrow = c(2, 1))
plot(marg.tau1, main = "tau1", type = "l")
plot(marg.tau2, main = "tau2", type = "l")

par(oldpar)

## Running UNIVARIATE MODEL

#Real data
d.uni <- list(OBS = nc.sids$SID74,
              NWPROP = nc.sids$NWPROP74,
              EXP = nc.sids$EXP74)
d.uni$idx <- 1:length(d.uni$OBS)

#Fit model
r.uni <- inla(OBS ~ 1 + f(idx, model = "besag", graph = W),
              data = d.uni, E = EXP, family = "poisson",
              control.predictor = list(compute = TRUE))
```

```
summary(r.uni)

nc.sids$FITTED74.uni <- r.uni$summary.fitted.values[ , "mean"]

#Display univariate VS multivariate fitted relative risks.

spplot(nc.sids, c("FITTED74", "FITTED74.uni"),
      main=list(label="Relative risk estimation",cex=2))

plot(nc.sids$FITTED74.uni, nc.sids$FITTED74, main="Relative Risk estimations",
    xlab="Univariate RR estimations"
    , ylab="Multivariate RR estimations", xlim=c(0.5, 2.5), ylim=c(0.5, 2.5))
abline(h=0, col="grey")
abline(v=0, col="grey")
abline(a=0, b=1, col="red")


#Plot posterior mean of the spatial effects univ VS multi

nc.sids$m.uni <- r.uni$summary.random$idx[, "mean"]
nc.sids$m.mult <- r$summary.random$idx[1:100, "mean"]

plot(nc.sids$m.uni, nc.sids$m.mult,
    main="Posterior mean of the spatial effect", xlab="Uni. post. means"
    , ylab="Mult. post. means", xlim=c(-1,1), ylim=c(-1,1))
abline(h=0, col="grey")
abline(v=0, col="grey")
abline(a=0, b=1, col="red")

spplot(nc.sids, c("m.mult", "m.uni"),
      main=list(label="Post. mean spatial effect",cex=2))

}
```

`inla.rgeneric.indep.MCAR.model` *$MCAR(\alpha, \Lambda)$: Proper multivariate CAR latent effect without correlation parameters.*

Description

Multivariate generalization of the proper conditional autorregresive model with a common spatial autocorrelation parameter. No correlation parameters are considered between the different diseases, so the matrix which models the variability between diseases will be a diagonal matrix.

Usage

```
inla.rgeneric.indep.MCAR.model(cmd, theta)
```

```
inla.INDMCAR.model(...)
```

Arguments

...	Arguments to be passed to 'inla.rgeneric.define'.
cmd	Arguments used by latent effects defined using the 'rgeneric' latent effect.
theta	Vector of hyperparameters.

Details

This function is used to define a latent effect that is a multivariate spatial effect with a proper conditional autorregresive distribution (with a common spatial autocorrelation parameter) and a diagonal matrix in order to model the whitin-disease and the between-diseases variability, respectively. Due to this effect is a multivariate spatial latent effect this function requires the following arguments when defining the latent effect:

- *W* Adjacency SPARSE matrix for spatial effect in the basic binary code.
- *k* Number of diseases of the multivariate study.
- *alpha.min* Minimum value of the spatial autocorrelation parameter.
- *alpha.max* Maximum value of the spatial autocorrelation parameter.

This model is defined using the 'f()' function and an index in order to identify the spatial areas. See the example.

Value

This is used internally by the 'INLA::inla()'.

Prior distributions of the hyperparameters

The hyperparamenters of this lattent effect are the marginal precisions of each disease and the common spatial autocorrelation parameter. So the total number of hyperpameters is equal to the number of diseases plus one.

References

Palmí-Perales F, Gómez-Rubio V, Martinez-Beneito MA (2021). "Bayesian Multivariate Spatial Models for Lattice Data with INLA." _Journal of Statistical Software_, *98*(2), 1-29. doi: 10.18637/jss.v098.i02 (URL: https://doi.org/10.18637/jss.v098.i02).

Examples

```
if (require("INLA", quietly = TRUE)) {
require(spdep)
require(spData)
require(rgdal)
```

```
# Load SIDS data
nc.sids <- readOGR(system.file("shapes/sids.shp", package="spData")[1])
proj4string(nc.sids) <- CRS("+proj=longlat +ellps=clrk66")

# Compute adjacency matrix, as nb object 'adj' and sparse matrix 'W'
adj <- poly2nb(nc.sids)
W <- as(nb2mat(adj, style = "B"), "Matrix")

# Compute expected cases
r74 <- sum(nc.sids$SID74) / sum(nc.sids$BIR74)
nc.sids$EXP74 <- r74 * nc.sids$BIR74
nc.sids$SMR74 <- nc.sids$SID74 / nc.sids$EXP74
nc.sids$NWPROP74 <- nc.sids$NWBIR74 / nc.sids$BIR74

r79 <- sum(nc.sids$SID79) / sum(nc.sids$BIR79)
nc.sids$EXP79 <- r79 * nc.sids$BIR79
nc.sids$SMR79 <- nc.sids$SID79 / nc.sids$EXP79
nc.sids$NWPROP79 <- nc.sids$NWBIR79 / nc.sids$BIR79

# Data (replicated to assess scalability)

#Real data
n.rep <- 1
d <- list(OBS = c(nc.sids$SID74, nc.sids$SID79),
         NWPROP = c(nc.sids$NWPROP74, nc.sids$NWPROP79),
         EXP = c(nc.sids$EXP74, nc.sids$EXP79))
d <- lapply(d, function(X) { rep(X, n.rep)})
d$idx <- 1:length(d$OBS)

# Model parameters
k <- 2 * n.rep #Number of diseases
alpha.min <- 0
alpha.max <- 1

#Define independent MCAR model
model <- inla.rgeneric.define(inla.rgeneric.indep.MCAR.model,
                              debug = FALSE, k = k, W = W,
                              alpha.min = alpha.min, alpha.max = alpha.max)


#Fit model
r <- inla(OBS ~ 1 + f(idx, model = model), # + NWPROP,
         data = d, E = EXP, family = "poisson",
         control.predictor = list(compute = TRUE))

summary(r)

# Transformed parameters
r.hyperpar <- inla.MCAR.transform(r, k = 2, model = "INDPMCAR",
  alpha.min = alpha.min, alpha.max = alpha.max)
r.hyperpar$summary.hyperpar

#Get fitted data, i.e., relative risk
```

```
nc.sids$FITTED74 <- r$summary.fitted.values[1:100, "mean"]
nc.sids$FITTED79 <- r$summary.fitted.values[100 + 1:100, "mean"]

#Display fitted relative risks
dev.new()
spplot(nc.sids, c("SMR74", "FITTED74", "SMR79", "FITTED79"))

# Showing results

#Show marginals of alpha, tau1, tau2
marg.alpha <- inla.tmarginal(
 function(x) alpha.min + (alpha.max - alpha.min) / (1 + exp(-x)),
 r$marginals.hyperpar[[1]])

marg.tau1 <- inla.tmarginal(
 function(x) exp(x),
 r$marginals.hyperpar[[2]])

marg.tau2 <- inla.tmarginal(
 function(x) exp(x),
 r$marginals.hyperpar[[3]])

dev.new()

oldpar <- par(mfrow = c(2, 2))

plot(marg.alpha, main="alpha", type="l")
plot(marg.tau1, main = "tau1", type = "l")
plot(marg.tau2, main = "tau2", type = "l")

par(oldpar)

## Running UNIVARIATE MODEL

#Real data
n.rep <- 1
d <- list(OBS = nc.sids$SID74,
         NWPROP = nc.sids$NWPROP74,
         EXP = nc.sids$EXP74)
d <- lapply(d, function(X) { rep(X, n.rep)})
d$idx <- 1:length(d$OBS)

#Fit model
r.uni <- inla(OBS ~ 1 + f(idx, model = "besag", graph = W), # + NWPROP,
             data = d, E = EXP, family = "poisson",
             control.predictor = list(compute = TRUE))

summary(r.uni)

nc.sids$FITTED74.uni <- r.uni$summary.fitted.values[ , "mean"]

#Display univariate VS multivariate  fitted relative risks.
dev.new()
```

```
spplot(nc.sids, c("SMR74", "FITTED74", "FITTED74.uni"))
spplot(nc.sids, c("FITTED74", "FITTED74.uni"),
      main=list(label="Relative risk estimation",cex=2))
dev.new()
plot(nc.sids$FITTED74.uni, nc.sids$FITTED74, main="Relative Risk estimations",
      xlab="Univariate RR estimations"
    , ylab="Multivariate RR estimations")#, xlim=c(0.5, 2.5), ylim=c(0, 2))
abline(h=0, col="grey")
abline(v=0, col="grey")
abline(a=0, b=1, col="red")

#Plot posterior mean of the spatial effects univ VS multi

nc.sids$m.uni <- r.uni$summary.random$idx[, "mean"]
nc.sids$m.mult <- r$summary.random$idx[1:100, "mean"]
dev.new()
plot(nc.sids$m.uni, nc.sids$m.mult,
    main="Posterior mean of the spatial effect", xlab="Uni. post. means"
    , ylab="Mult. post. means")#, xlim=c(-1,1), ylim=c(-7,1))
abline(h=0, col="grey")
abline(v=0, col="grey")
abline(a=0, b=1, col="red")

dev.new()
spplot(nc.sids, c("m.mult", "m.uni"),
      main=list(label="Post. mean spatial effect",cex=2))
}
```

`inla.rgeneric.MCAR.model` *$MCAR(\alpha, \Lambda)$: Proper multivariate CAR latent effect with a common autorcorrelation parameter.*

Description

Multivariate generalization of the proper conditional autorregresive model with one common correlation parameter. The matrix which models the variability between diseases is a symmetric matrix with the inverse of the marginal precisions on the diagonal elements and the correlation parameters divided by the square root of the precisions on the off-diagonal elements.

Usage

```
inla.rgeneric.MCAR.model(cmd, theta)

inla.MCAR.model(...)
```

Arguments

...	Arguments to be passed to 'inla.rgeneric.define'.
cmd	Arguments used by latent effects defined using the 'rgeneric' latent effect.
theta	Vector of hyperparameters.

Details

This function is used to define a latent effect that is a multivariate spatial effect with a proper conditional autorregresive distribution (with a common spatial autocorrelation parameter) and a symmetric matrix in order to model the whitin-disease and the between-diseases variability, respectively. Due to this effect is a multivariate spatial latent effect this function requires the following arguments when defining the latent effect:

- *W* Adjacency SPARSE matrix for spatial effect in the basic binary code.
- *k* Number of diseases of the multivariate study.
- *alpha.min* Minimum value of the spatial autocorrelation parameter.
- *alpha.max* Maximum value of the spatial autocorrelation parameter.

This model is defined using the 'f()' function and an index in order to identify the spatial areas. See the example.

Value

This is used internally by the 'INLA::inla()'.

Prior distributions of the hyperparameters

The hyperparamenters of this lattent effect are the marginal precisions of each disease which are equal to the number of diseases, the correlation parameters for the whole pair of diseases and the common spatial autocorrelation parameter.

References

Palmí-Perales F, Gómez-Rubio V, Martinez-Beneito MA (2021). "Bayesian Multivariate Spatial Models for Lattice Data with INLA." _Journal of Statistical Software_, *98*(2), 1-29. doi: 10.18637/jss.v098.i02 (URL: https://doi.org/10.18637/jss.v098.i02).

Examples

```
if (require("INLA", quietly = TRUE)) {
require(spdep)
require(spData)
require(rgdal)

#Load SIDS data
nc.sids <- readOGR(system.file("shapes/sids.shp", package="spData")[1])
proj4string(nc.sids) <- CRS("+proj=longlat +ellps=clrk66")
```

```
#Compute adjacency matrix, as nb object 'adj' and sparse matrix 'W'
adj <- poly2nb(nc.sids)
W <- as(nb2mat(adj, style = "B"), "Matrix")

#Compute expected cases
r74 <- sum(nc.sids$SID74) / sum(nc.sids$BIR74)
nc.sids$EXP74 <- r74 * nc.sids$BIR74
nc.sids$SMR74 <- nc.sids$SID74 / nc.sids$EXP74
nc.sids$NWPROP74 <- nc.sids$NWBIR74 / nc.sids$BIR74

r79 <- sum(nc.sids$SID79) / sum(nc.sids$BIR79)
nc.sids$EXP79 <- r79 * nc.sids$BIR79
nc.sids$SMR79 <- nc.sids$SID79 / nc.sids$EXP79
nc.sids$NWPROP79 <- nc.sids$NWBIR79 / nc.sids$BIR79

# Data (replicated to assess scalability)

#Real data
n.rep <- 1
d <- list(OBS = c(nc.sids$SID74, nc.sids$SID79),
          NWPROP = c(nc.sids$NWPROP74, nc.sids$NWPROP79),
          EXP = c(nc.sids$EXP74, nc.sids$EXP79))
d <- lapply(d, function(X) { rep(X, n.rep)})
d$idx <- 1:length(d$OBS)

# Model parameters
k <- 2 * n.rep #Number of diseases
alpha.min <- 0
alpha.max <- 1

#Define MCAR model
#model <- inla.rgeneric.define(inla.rgeneric.MCAR.model, debug = FALSE,
#  k = k, W = W, alpha.min = alpha.min, alpha.max = alpha.max)
model <- inla.MCAR.model(k = k, W = W, alpha.min = alpha.min, alpha.max = alpha.max)


#Fit model
r <- inla(OBS ~ 1 + f(idx, model = model),
  data = d, E = EXP, family = "poisson",
  control.compute = list(config = TRUE),
  control.predictor = list(compute = TRUE))

summary(r)

# Transformed parameters
r.hyperpar <- inla.MCAR.transform(r, k = 2, model = "PMCAR",
  alpha.min = alpha.min, alpha.max = alpha.max)
r.hyperpar$summary.hyperpar

#Get fitted data, i.e., relative risk
nc.sids$FITTED74 <- r$summary.fitted.values[1:100, "mean"]
nc.sids$FITTED79 <- r$summary.fitted.values[100 + 1:100, "mean"]
```

```
#Display fitted relative risks
dev.new()
spplot(nc.sids, c("SMR74", "FITTED74", "SMR79", "FITTED79"))


# Showing results of the MCAR: multivariate proper CAR.

#Show marginals of alpha, tau1, tau2
marg.alpha <- inla.tmarginal(
  function(x) alpha.min + (alpha.max - alpha.min) / (1 + exp(-x)),
  r$marginals.hyperpar[[1]])

marg.tau1 <- inla.tmarginal(
  function(x) exp(x),
  r$marginals.hyperpar[[2]])

marg.tau2 <- inla.tmarginal(
  function(x) exp(x),
  r$marginals.hyperpar[[3]])

dev.new()

oldpar <- par(mfrow = c(2, 2))

plot(marg.alpha, main="alpha", type="l")
plot(marg.tau1, main = "tau1", type = "l")
plot(marg.tau2, main = "tau2", type = "l")

par(oldpar)

## Running UNIVARIATE MODEL

#Real data
n.rep <- 1
d <- list(OBS = nc.sids$SID74,
          NWPROP = nc.sids$NWPROP74,
          EXP = nc.sids$EXP74)
d <- lapply(d, function(X) { rep(X, n.rep)})
d$idx <- 1:length(d$OBS)

#Fit model
r.uni <- inla(OBS ~ 1 + f(idx, model = "besag", graph = W),
              data = d, E = EXP, family = "poisson",
              control.predictor = list(compute = TRUE))

summary(r.uni)

nc.sids$FITTED74.uni <- r.uni$summary.fitted.values[ , "mean"]

#Display univariate VS multivariate  fitted relative risks.
dev.new()
spplot(nc.sids, c("SMR74", "FITTED74", "FITTED74.uni"))
```

```
spplot(nc.sids, c("FITTED74", "FITTED74.uni"),
       main=list(label="Relative risk estimation",cex=2))
dev.new()
plot(nc.sids$FITTED74.uni, nc.sids$FITTED74,
     main="Relative Risk estimations", xlab="Univariate RR estimations"
     , ylab="Multivariate RR estimations")
abline(h=0, col="grey")
abline(v=0, col="grey")
abline(a=0, b=1, col="red")

#Plot posterior mean of the spatial effects univ VS multi

nc.sids$m.uni <- r.uni$summary.random$idx[, "mean"]
nc.sids$m.mult <- r$summary.random$idx[1:100, "mean"]
dev.new()
plot(nc.sids$m.uni, nc.sids$m.mult,
     main="Posterior mean of the spatial effect",
     xlab="Uni. post. means", ylab="Mult. post. means")
abline(h=0, col="grey")
abline(v=0, col="grey")
abline(a=0, b=1, col="red")

dev.new()
spplot(nc.sids, c("m.mult", "m.uni"),
       main=list(label="Post. mean spatial effect",cex=2))
}
```

`inla.rgeneric.Mmodel.model` *M-model: Proper multivariate CAR latent effect with a different spatial autorcorrelation parameter for each disease.*

Description

Multivariate generalization of the proper conditional autorregresive model with one common correlation parameter. This model is performed using the M-model aproximation of Rocamora et. al. (2015).

Usage

```
inla.rgeneric.Mmodel.model(cmd, theta)

inla.Mmodel.model(...)
```

Arguments

...	Arguments to be passed to 'inla.rgeneric.define'.

cmd	Arguments used by latent effects defined using the 'rgeneric' latent effect.
theta	Vector of hyperparameters.

Details

This function is used to define a latent effect that is a multivariate spatial effect based on the M-model aproximation of Rocamora et. al. (2015) in which θ is modelled as a product of a $\Phi \cdot M$ where the colums of Φ are modeled independently with a proper conditional autorregresive distribution with a different spatial autocorrelation parameter for each disease and M is a square matrix which introduce de dependence between the diseases. Due to this effect is a multivariate spatial latent effect this function requires the following arguments when defining the latent effect:

- *W* Adjacency SPARSE matrix for spatial effect in the basic binary code.
- *k* Number of diseases of the multivariate study.
- *alpha.min* Minimum value of the spatial autocorrelation parameter.
- *alpha.max* Maximum value of the spatial autocorrelation parameter.

This model is defined using the 'f()' function and an index in order to identify the spatial areas. See the example.

Value

This is used internally by the 'INLA::inla()'.

Prior distributions of the hyperparameters

The hyperparamenters of this lattent effect are the common spatial autocorrelation parameters (one for each disease) and the entries of the M matrix (considered all as a random effects).

References

Palmí-Perales F, Gómez-Rubio V, Martinez-Beneito MA (2021). "Bayesian Multivariate Spatial Models for Lattice Data with INLA." _Journal of Statistical Software_, *98*(2), 1-29. doi: 10.18637/jss.v098.i02 (URL: https://doi.org/10.18637/jss.v098.i02).

Examples

```
if (require("INLA", quietly = TRUE)) {
require(spdep)
require(spData)
require(rgdal)

#Load SIDS data
nc.sids <- readOGR(system.file("shapes/sids.shp", package="spData")[1])
proj4string(nc.sids) <- CRS("+proj=longlat +ellps=clrk66")

#Compute adjacency matrix, as nb object 'adj' and sparse matrix 'W'
adj <- poly2nb(nc.sids)
```

```
W <- as(nb2mat(adj, style = "B"), "Matrix")

#Compute expected cases
r74 <- sum(nc.sids$SID74) / sum(nc.sids$BIR74)
nc.sids$EXP74 <- r74 * nc.sids$BIR74
nc.sids$SMR74 <- nc.sids$SID74 / nc.sids$EXP74
nc.sids$NWPROP74 <- nc.sids$NWBIR74 / nc.sids$BIR74

r79 <- sum(nc.sids$SID79) / sum(nc.sids$BIR79)
nc.sids$EXP79 <- r79 * nc.sids$BIR79
nc.sids$SMR79 <- nc.sids$SID79 / nc.sids$EXP79
nc.sids$NWPROP79 <- nc.sids$NWBIR79 / nc.sids$BIR79

# Data (replicated to assess scalability)

#Real data
n.rep <- 1
d <- list(OBS = c(nc.sids$SID74, nc.sids$SID79),
          NWPROP = c(nc.sids$NWPROP74, nc.sids$NWPROP79),
          EXP = c(nc.sids$EXP74, nc.sids$EXP79))
d <- lapply(d, function(X) { rep(X, n.rep)})
d$idx <- 1:length(d$OBS)

#Parameters of the Mmodel
k <- 2
alpha.min <- 0
alpha.max <- 1



model <- inla.rgeneric.define(inla.rgeneric.Mmodel.model, debug = FALSE,
                              k = k, W = W, alpha.min = alpha.min,
                              alpha.max = alpha.max)

r.Mmodel <- inla(OBS ~ -1 + f(idx, model = model), data = d, E = EXP,
  family = "poisson", control.predictor = list(compute = TRUE))

nc.sids$Model1 <- r.Mmodel$summary.random$idx[1:100, "mean"]
nc.sids$Model2 <- r.Mmodel$summary.random$idx[100 + 1:100, "mean"]

spplot(nc.sids, c("Model1", "Model2"))

nc.sids$Fit1 <- r.Mmodel$summary.fitted[1:100, "mean"]
nc.sids$Fit2 <- r.Mmodel$summary.fitted[100 + 1:100, "mean"]

spplot(nc.sids, c("Fit1", "SMR74", "Fit2", "SMR79"))


## Running UNIVARIATE MODEL

#Real data
n.rep <- 1
d <- list(OBS = nc.sids$SID74,
```

```
          NWPROP = nc.sids$NWPROP74,
          EXP = nc.sids$EXP74)
d <- lapply(d, function(X) { rep(X, n.rep)})
d$idx <- 1:length(d$OBS)

#Fit model
r.uni <- inla(OBS ~ 1 + f(idx, model = "besag", graph = W), # + NWPROP,
              data = d, E = EXP, family = "poisson",
              control.predictor = list(compute = TRUE))

summary(r.uni)

nc.sids$FITTED74.uni <- r.uni$summary.fitted.values[ , "mean"]

#Display univariate VS multivariate  fitted relative risks.
dev.new()
spplot(nc.sids, c("SMR74", "Fit1", "FITTED74.uni"))
spplot(nc.sids, c("Fit1", "FITTED74.uni"),
       main=list(label="Relative risk estimation",cex=2))
dev.new()
plot(nc.sids$FITTED74.uni, nc.sids$Fit1, main="Relative Risk estimations",
     xlab="Univariate RR estimations"
     , ylab="Multivariate RR estimations")#, xlim=c(0.5, 2.5), ylim=c(0, 2))
abline(h=0, col="grey")
abline(v=0, col="grey")
abline(a=0, b=1, col="red")

}
```

Index