

Spatial lag model for spatial effects

Parametrization

The `slm` model is defined as

$$\mathbf{x} = (I_n - \rho W)^{-1}(X\beta + \varepsilon)$$

where I_n is the identity matrix of dimension $n \times n$, W is a spatial weights matrix, X is a matrix of covariates, ρ is a spatial autocorrelation parameter, β are coefficients of the covariates and ε is zero mean Gaussian noise with precision τ .

Hyperparameters

This model has two hyperparameters $\theta = (\theta_1, \theta_2)$. The precision parameter τ is represented as

$$\theta_1 = \log \tau$$

and the prior is defined on θ_1 . The spatial autocorrelation parameter ρ is represented as

$$\rho^* = \frac{\rho - \rho_{\min}}{\rho_{\max} - \rho_{\min}}$$

and then

$$\theta_2 = \log(\rho^*/(1 - \rho^*))$$

and the prior is defined on θ_2 . Here, ρ_{\min} and ρ_{\max} are lower and upper limits of the legal range for ρ .

Specification

The `slm` model is specified inside the `f()` function as

```
f(<whatever>, model="slm",
  args.slm = list(rho.min = NULL,
                 rho.max = NULL,
                 X = NULL,
                 W = NULL,
                 Q.beta = NULL))
```

`args.slm` is used to define the `slm`-specific parameters in the model.

rho.min and **rho.max** define the range in which ρ can take values. Note that, ρ^* is in the interval $(0, 1)$ and that it is re-scaled to the interval **(rho.min, rho.max)** when computing $I_n - \rho W$. Initial values on ρ need to be re-scaled to the $(0, 1)$ interval.

X defines the matrix of covariates.

W defines the adjacency matrix.

Q.beta defines the precision of the vector of coefficients β in the model.

Hyperparameter specification and default values

hyper

theta1

hyperid 34001
name log precision
short.name prec
initial 4
fixed FALSE
prior loggamma
param 1 5e-05
to.theta function(x) log(x)
from.theta function(x) exp(x)

theta2

hyperid 34002
name rho
short.name rho
initial 0
fixed FALSE
prior normal
param 0 10
to.theta function(x) log(x/(1-x))
from.theta function(x) 1/(1+exp(-x))

constr FALSE

nrow.ncol FALSE

augmented FALSE

aug.factor 1

aug.constr

n.div.by

n.required TRUE

set.default.values TRUE

pdf slm

status experimental

Example

Example using the Boston dataset from package spdep

```
require(INLA)
require(spdep)
data(boston)
```

```

## Index for the latent model
n <- nrow(boston.c)
boston.c$idx <- 1:n

## Define adjacency using a row-standardised matrix
lw <- nb2listw(boston.soi)
W <- as(as_dgRMatrix_listw(lw), "CsparseMatrix")

## Model definition
f1 <- log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2) + AGE +
  log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT)
mmatrix <- model.matrix(f1, boston.c)

## Zero-variance for error term
zero.variance = list(prec=list(initial = 25, fixed=TRUE))

## Compute eigenvalues for SLM model, used to obtain rho.min and
## rho.max
e = eigenw(lw)
re.idx = which(abs(Im(e)) < 1e-6)
rho.max = 1/max(Re(e[re.idx]))
rho.min = 1/min(Re(e[re.idx]))
rho = mean(c(rho.min, rho.max))

## Precision matrix for beta coefficients' prior
betaprec <- .0001
Q.beta = Diagonal(n=ncol(mmatrix), betaprec)

## Priors on the hyperparameters
hyper = list(
  prec = list(
    prior = "loggamma",
    param = c(0.01, 0.01)),
  rho = list(
    initial=0,
    prior = "logitbeta",
    param = c(1,1)))

## Fit model
slmm1 <- inla( log(CMEDV) ~ -1 +
  f(idx, model="slm",
    args.slm=list(
      rho.min = rho.min,
      rho.max = rho.max,
      W=W,
      X=mmatrix,
      Q.beta=Q.beta),
    hyper=hyper),
  data=boston.c, family="gaussian",
  control.family = list(hyper=zero.variance),
  control.compute=list(dic=TRUE, cpo=TRUE)
)
summary(slmm1)

## Summary of the coefficients (at the end of the vector of random effects)
slmm1$summary.random$idx[n+1:ncol(mmatrix),]

```

```
## Re-scale rho to real scale
rhomarg <- inla.tmarginal(function(x){rho.min+x*(rho.max-rho.min)},
                          slmm1$marginals.hyperpar[[2]])
inla.zmarginal(rhomarg)

## Maximum likelihood estimate of model (used for comparison)
summary(m2 <- lagsarlm(f1, boston.c, lw))
```

Notes

The estimates of β are included at the end of the vector of random effects. See the example for details on how to extract them.