Package ‘ref.ICAR’

November 19, 2018

Title Objective Bayes Intrinsic Conditional Autoregressive Model for Areal Data

Version 1.0

Author Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Maintainer Erica M. Porter <ericamp@vt.edu>

Depends R (>= 3.1.0)

Description Implements an objective Bayes intrinsic conditional autoregressive prior. This model provides an objective Bayesian approach for modeling spatially correlated areal data using an intrinsic conditional autoregressive prior on a vector of spatial random effects.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports rgdal, spdep, mvtnorm, coda, MCMCglmm, Rdpack, graphics

RdMacros Rdpack

Suggests maptools, maps, MASS, sp, knitr, rmarkdown, RColorBrewer, captioner, rcrossref

VignetteBuilder knitr

BuildManual yes

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-19 18:00:06 UTC

R topics documented:

ref.ICAR-package .................................................. 2
ref.analysis ......................................................... 3
ref.MCMC .......................................................... 4
**Description**

Implements the objective prior for intrinsic conditional autoregressive (ICAR) random effects proposed by Keefe et al. (2018). This model provides an objective Bayesian approach for modeling spatially correlated areal data using an ICAR prior on a vector of spatial random effects.

**Details**

- **Package:** ref.ICAR
- **Type:** Package
- **Version:** 1.0
- **Date:** 2018-10-10
- **License:** GPL (>= 2)

**Author(s)**

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Maintainer: Erica M. Porter <ericamp@vt.edu>

**References**


**Examples**

```r
## Refer to the vignette attached to the package,
## and to selected help files.
```
MCMC Analysis and Summaries for Reference Prior on an Intrinsic Autoregressive Model for Areal Data

Description

Performs analysis on a geographical areal data set using the objective prior for intrinsic conditional autoregressive (ICAR) random effects (Keefe et al. 2018). It takes a shapefile, data, and region names to construct a neighborhood matrix and perform Markov chain Monte Carlo sampling on the unstructured and spatial random effects. Finally, the function obtains regional estimates and performs posterior inference on the model parameters.

Usage

ref.analysis(shape.file, X, y, x.reg.names, y.reg.names, shp.reg.names = NULL, iters = 10000, burnin = 5000, verbose = TRUE, tau.c.start = 1, beta.start = 1, sigma2.start = 1, step.tauc = 0.5, step.sigma2 = 0.5)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>shape.file</td>
<td>A shapefile corresponding to the regions for analysis.</td>
</tr>
<tr>
<td>X</td>
<td>A matrix of covariates, which should include a column of 1’s for models with a non-zero intercept</td>
</tr>
<tr>
<td>y</td>
<td>A vector of responses.</td>
</tr>
<tr>
<td>x.reg.names</td>
<td>A vector specifying the order of region names contained in X.</td>
</tr>
<tr>
<td>y.reg.names</td>
<td>A vector specifying the order of region names contained in y.</td>
</tr>
<tr>
<td>shp.reg.names</td>
<td>A vector specifying the order of region names contained in the shapefile, if there is not a NAME column in the file.</td>
</tr>
<tr>
<td>iters</td>
<td>Number of MCMC iterations to perform. Defaults to 10,000.</td>
</tr>
<tr>
<td>burnin</td>
<td>Number of MCMC iterations to discard as burn-in. Defaults to 5,000.</td>
</tr>
<tr>
<td>verbose</td>
<td>If FALSE, MCMC progress is not printed.</td>
</tr>
<tr>
<td>tau.c.start</td>
<td>Starting MCMC value for the spatial dependence parameter.</td>
</tr>
<tr>
<td>beta.start</td>
<td>Starting MCMC value for the fixed effect regression coefficients.</td>
</tr>
<tr>
<td>sigma2.start</td>
<td>Starting MCMC value for the variance of the unstructured random effects.</td>
</tr>
<tr>
<td>step.tauc</td>
<td>Step size for the spatial dependence parameter.</td>
</tr>
<tr>
<td>step.sigma2</td>
<td>Step size for the variance of the unstructured random effects.</td>
</tr>
</tbody>
</table>
Value

A list containing \( h \), MCMC chains, parameter summaries, fitted regional values, and regional summaries.

\( h \) The neighborhood matrix.

MCMC Matrix of MCMC chains for all model parameters.

beta.median Posterior medians of the fixed effect regression coefficients.

beta.hpd Highest Posterior Density intervals for the fixed effect regression coefficients.

tauc.median Posterior median of the spatial dependence parameter.

tauc.hpd Highest Posterior Density interval for the spatial dependence parameter.

sigma2.median Posterior median of the unstructured random effects variance.

sigma2.hpd Highest Posterior Density interval for the unstructured random effects variance.

tauc.accept Final acceptance rate for the spatial dependence parameter.

sigma2.accept Final acceptance rate for the unstructured random effects variance.

fit.dist Matrix of fitted posterior values for each region in the data.

reg.medians Vector of posterior medians for fitted response by region.

reg.hpd Data frame of Highest Posterior Density intervals by region.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

```r
## Refer to the vignette attached to the package.
```

Description

Implements the Metropolis-within-Gibbs sampling algorithm proposed by Keefe et al. (2018), to perform posterior inference for the intrinsic conditional autoregressive model with spatial random effects.

Usage

```r
ref.MCMC(y, X, H, iters = 10000, burnin = 5000, verbose = TRUE, 
  tauc.start = 1, beta.start = 1, sigma2.start = 1, 
  step.tauc = 0.5, step.sigma2 = 0.5)
```
Arguments

- `y` Vector of responses.
- `X` Matrix of covariates. This should include a column of 1’s for models with a non-zero intercept.
- `H` The neighborhood matrix.
- `iters` Number of MCMC iterations to perform. Defaults to 10,000.
- `burnin` Number of MCMC iterations to discard as burn-in. Defaults to 5,000.
- `verbose` If FALSE, MCMC progress is not printed.
- `tauc.start` Starting value for the spatial dependence parameter.
- `beta.start` Starting value for the vector of fixed effect regression coefficients.
- `sigma2.start` Starting value for the variance of the unstructured random effects.
- `step.tauc` Step size for the spatial dependence parameter
- `step.sigma2` Step size for the variance of the unstructured random effects.

Value

A list containing MCMC chains and parameter summaries.

- `MCMCchain` Matrix of MCMC chains.
- `tauc.MCMC` MCMC chains for the spatial dependence parameter.
- `sigma2.MCMC` MCMC chains for the variance of the unstructured random effects.
- `phi.MCMC` MCMC chains for the spatial random effects.
- `beta.MCMC` MCMC chains for the fixed effect regression coefficients.
- `accept.sigma2` Final acceptance number for variance of the unstructured random effects.
- `accept.tauc` Final acceptance number for spatial dependence parameter.
- `accept.phi` Final acceptance number for spatial random effects.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

References


Examples

##### Fit the model for simulated areal data on a grid #####

##### Load extra libraries
library(sp)
library(methods)
library(spdep)
library(mvtnorm)

##### Generate areal data on a grid
rows=5; cols=5
tauc=1
sigma2=2; beta=c(1,5)

##### Create grid
grid <- GridTopology(c(1,1), c(1,1), c(cols,rows))
polys <- as(grid, ”SpatialPolygons”)
spgrid <- SpatialPolygonsDataFrame(polys, data=data.frame(row.names=row.names(polys)))

##### Create neighborhood matrix
grid.nb <- poly2nb(spgrid, queen=FALSE)
W <- nb2mat(grid.nb, style=”B”)

##### Put spatially correlated data in grid
p <- length(beta)
um.reg <- (rows*cols)
if(p>1){x1<-rmvnorm(n=num.reg, mean=rep(0,p-1), sigma=diag(p-1))} else{x1<-NULL}
X <- cbind(rep(1,num.reg), x1)
Dmat <- diag(apply(W,1,sum))
H <- Dmat - W
row.names(H) <- NULL

##### Obtain true response vector
theta_true <- rnorm(num.reg, mean=0, sd=sqrt(sigma2))
Q <- eigen(H,symmetric=TRUE)$vectors
eigh <- eigen(H,symmetric=TRUE)$values
D <- diag(eigh)
Qmat <- Q[,1:(num.reg-1)]
phimat <- diag(1/sqrt(eigh[1:(num.reg-1)]))
z <- t(rmvnorm(1,mean=rep(0,num.reg-1),sigma=diag(num.reg-1)))
phi_true <- sqrt((1/tauc)*sigma2)*(Qmat%*%phimat%*%z)
Y <- X%*%beta + theta_true + phi_true

##### Fit the model
set.seed(5432)
model <- ref.MCMC(y=Y, X=X, H=H, iters=15000, burnin=5000, verbose=TRUE, tauc.start=.1, beta.start=-1, sigma2.start=.1, step.tau=0.5, step.sigma2=0.5)

##### Small example for checking
model <- ref.MCMC(y=Y, X=X, H=H, iters=1000, burnin=50, verbose=TRUE, tauc.start=.1, beta.start=-1,
Description

This function creates trace plots for the parameters in the ICAR reference prior model (Keefe et al. 2018).

Usage

ref.plot(MCMCchain, X, burnin, num.reg)

Arguments

MCMCchain    Matrix of MCMC chains for the model parameters.
X             Matrix of covariates.
burnin       Number of MCMC iterations from MCMCchain discarded as burn-in.
um.reg       Number of regions in the areal data set.

Value

Trace plots for the fixed effect regression coefficients, the precision parameter, and the unstructured random effects variance.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

## Refer to the vignette attached to the package.
Parameter Summaries for MCMC Analysis

Description

Takes a matrix of MCMC chains, iterations, and acceptance values to return posterior summaries of the parameters, including posterior medians, intervals, and acceptance rates.

Usage

ref.summary(MCMCchain, tauc.MCMC, sigma2.MCMC, beta.MCMC, phi.MCMC, accept.phi, accept.sigma2, accept.tauc, iters = 10000, burnin = 5000)

Arguments

MCMCchain: Matrix of MCMC chains for the ICAR model parameters.
tauc.MCMC: MCMC chains for the spatial dependence parameter.
sigma2.MCMC: MCMC chains for the variance of the unstructured random effects.
beta.MCMC: MCMC chains for the fixed effect regression coefficients.
phi.MCMC: MCMC chains for the spatial random effects.
accept phi: Final acceptance number for spatial random effects.
accept sigma2: Final acceptance number for variance of the unstructured random effects.
accept tau: Final acceptance number for the spatial dependence parameter.
iters: Number of MCMC iterations in MCMCchain.
burnin: Number of MCMC iterations discarded as burn-in for MCMCchain.

Value

Parameter summaries

beta.median: Posterior medians of the fixed effect regression coefficients.
beta.hpd: Highest Posterior Density intervals for the fixed effect regression coefficients.
tauc.median: Posterior median of the spatial dependence parameter.
tauc.hpd: Highest Posterior Density interval for the spatial dependence parameter.
sigma2.median: Posterior median of the unstructured random effects variance.
sigma2.hpd: Highest Posterior Density interval for the unstructured random effects variance.
accept tau: Final acceptance rate for the spatial dependence parameter.
sigma2.accept: Final acceptance rate for the unstructured random effects variance.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

## Refer to the vignette attached to the package.
reg.summary

Regional Summaries for Areal Data Modeled by ICAR Reference Prior Model

Description

This function takes data and sampled MCMC chains for an areal data set and gives fitted posterior values and summaries by region using the model by (Keefe et al. 2018).

Usage

reg.summary(MCMCchain, X, Y, burnin)

Arguments

MCMCchain       Matrix of MCMC chains, using the sampling from (Keefe et al. 2018).
X                Matrix of covariates.
Y                Vector of responses.
burnin          Number of MCMC iterations discarded as burn-in in MCMCchain.

Value

A list of the fitted distributions by region, and medians and credible intervals by region.

fit.dist         Matrix of fitted posterior values for each region in the data.
reg.medians      Vector of posterior medians for fitted response by region.
reg.cred         Data frame of credible intervals by region.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

## Refer to the vignette attached to the package.
Creating a Neighborhood Matrix for Areal Data from a Shapefile

Description

Takes a path to a shape file and creates a neighborhood matrix. This neighborhood matrix can be used with the objective ICAR model (Keefe et al. 2018).

Usage

shapeNh(shapefile)

Arguments

shapefile: File path to a shapefile.

Value

A list containing a neighborhood matrix and the SpatialPolygonsDataFrame object corresponding to the shape file.

H: A neighborhood matrix.
map: SpatialPolygonsDataFrame object from the provided shapefile.

Author(s)

Erica M. Porter, Matthew J. Keefe, Christopher T. Franck, and Marco A.R. Ferreira

Examples

# Load extra libraries
library(sp)
library(rgdal)

# Read in a shapefile of the contiguous U.S. from package data
system.path <- system.file("extdata", "us.shape48.shp", package = "ref.ICAR", mustWork = TRUE)
shp.layer <- gsub('.shp', '', basename(system.path))
shp.path <- dirname(system.path)
us.shape48 <- readOGR(dsn = path.expand(shp.path), layer = shp.layer, verbose = FALSE)

shp.data <- shapeH(system.path)
names(shp.data)
Index

ref.analysis, 3
ref.ICAR(ref.ICAR-package), 2
ref.ICAR-package, 2
ref.MCMC, 4
ref.plot, 7
ref.summary, 8
reg.summary, 9

shape.H, 10