

Package ‘CARBayes’

January 27, 2015

Type Package

Title Spatial Areal Unit Modelling

Version 4.0

Date 2014-12-15

Author Duncan Lee

Maintainer Duncan Lee <Duncan.Lee@glasgow.ac.uk>

Description Implements Bayesian hierarchical spatial areal unit models. In such models the spatial autocorrelation is modelled by a set of random effects, which are assigned a conditional autoregressive (CAR) prior distribution. Examples of the models included are the BYM model as well as a recently developed localised spatial smoothing model. The initial creation of this package was supported by the Economic and Social Research Council (ESRC) grant RES-000-22-4256, and on-going development is supported by the Engineering and Physical Science Research Council (EPSRC) grant EP/J017442/1.

License GPL (>= 2)

Depends R (>= 2.10.0), MASS, Rcpp (>= 0.10.6)

Imports truncdist, spam, coda, sp

Suggests boot, deldir, foreign, grid, maptools, Matrix, nlme, shapefiles, spdep, splines

LinkingTo Rcpp

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-12-16 17:35:40

R topics documented:

| | |
|----------------------------------|---|
| CARBayes-package | 2 |
| combine.data.shapefile | 4 |
| highlight.borders | 5 |
| print.carbayes | 6 |
| S.CARbym | 7 |

| | |
|------------------------------|----|
| S.CARcluster | 9 |
| S.CARdissimilarity | 13 |
| S.CARiar | 16 |
| S.CARleroux | 18 |
| S.independent | 21 |
| spatialhousedata | 23 |
| spatialrespdata | 24 |
| summarise.lincomb | 25 |
| summarise.samples | 26 |

| | |
|--------------|-----------|
| Index | 27 |
|--------------|-----------|

| | |
|------------------|-------------------------------------|
| CARBayes-package | <i>Spatial Areal Unit Modelling</i> |
|------------------|-------------------------------------|

Description

Implements Bayesian hierarchical spatial areal unit models. In such models the spatial correlation is modelled by a set of random effects, which are assigned a conditional autoregressive (CAR) prior distribution. Examples of the models included are the BYM model as well as a recently developed localised spatial smoothing model. In addition, the package includes functions to enable plotting spatial maps of data (by combining the data with an appropriate shapefile) using the functionality of the sp package. There is a vignette accompanying this package. The initial creation of this package was supported by the Economic and Social Research Council (ESRC) grant RES-000-22-4256, and on-going development is supported by the Engineering and Physical Science Research Council (EPSRC) grant EP/J017442/1.

Details

Package: CARBayes
 Type: Package
 Version: 4.0
 Date: 2014-12-15
 License: GPL (>= 2)

Implements Bayesian hierarchical models for spatial areal unit data. The models are an extension of generalised linear models, and the response variable can be binomial, Gaussian or Poisson. The mean function for the data comprises a covariate component and a set of random effects. This package implements a number of different random effects models, which represent different types of spatial autocorrelation structure. These models are within the conditional autoregressive (CAR) class, and capture spatial correlation via a neighbourhood matrix W . This matrix typically (although other options are allowed) has a jk th element equal to one and zero, depending on whether areas j and k are spatially close (typically chosen to mean sharing a common border). Inference for all models is via Markov Chain Monte Carlo (MCMC) simulation. The following random effects models are implemented in this package.

Independent and identically distributed random effects.

Spatially correlated random effects modelled by the intrinsic autoregressive (IAR) model proposed by Besag et al (1991).

Spatially correlated random effects modelled by the Besag-York-Mollie (BYM) model proposed by Besag et al (1991).

Spatially correlated random effects modelled by the proper CAR model proposed by Leroux (1999).

Spatially correlated random effects modelled by the localised spatial smoothing approach proposed by Lee and Mitchell (2012).

Spatially correlated random effects modelled by the localised spatial smoothing approach proposed by Lee and Sarran (2014).

More detailed descriptions of the methods are given by the references below. CARBayes has several changes from version 3, one of which is the availability of an up-to-date vignette. Other changes are the removal of the proper CAR model by Stern et al (1999), and a change in the notation of the main functions from, for example, `lerouxCAR.re()` to `S.CARleroux()`. Finally, two additional functions have been added, which allow the posterior samples (`summarise.samples()`) and linear combinations of the covariate component (`summarise.lincomb()`), for example to summarise non-linear relationships, to be summarised.

Author(s)

Maintainer: Duncan Lee <Duncan.Lee@glasgow.ac.uk>

References

Besag, J. and York, J and Mollie, A (1991). Bayesian image restoration with two applications in spatial statistics. *Annals of the Institute of Statistics and Mathematics* 43, 1-59.

Eddelbuettel, D and Francois, R (2011). Rcpp: seamless R and C++ Integration. *Journal of Statistical Software*, 40, 1-18.

Stern, H and Cressie, N. (1999). Inference for extremes in disease mapping, Chapter Disease mapping and Risk Assessment for Public Health. Lawson, A and Biggeri, D and Boehning, E and Lesaffre, E and Viel, J and Bertollini, R (eds). Wiley.

Lee, D. (2011). A comparison of conditional autoregressive model used in Bayesian disease mapping. *Spatial and Spatio-temporal Epidemiology* 2, 79-89.

Lee, D. and Mitchell, R (2012). Boundary detection in disease mapping studies. *Biostatistics*, 13, 415-426.

Lee, D (2013). CARBayes: An R Package for Bayesian Spatial Modeling with Conditional Autoregressive Priors. *Journal of Statistical Software*, 55, 13, 1-24.

Leroux, B., Lei, X and Breslow, N (1999). Estimation of disease rates in small areas: A new mixed model for spatial dependence, Chapter Statistical Models in Epidemiology, the Environment and Clinical Trials, Halloran, M and Berry, D (eds), pp. 135-178. Springer-Verlag, New York.

Lee, D and Sarran, C (2014). Controlling for unmeasured confounding and spatial misalignment in long-term air pollution and health studies. [arXiv:1412.4479](https://arxiv.org/abs/1412.4479).

Examples

```

library(shapefiles)
library(sp)
library(maptools)
library(splines)
library(spdep)
library(CARBayes)

#### Read in the data
data(spatialhousedata)
housedata <- spatialhousedata@data

#### Transform the price and driveshop variables
housedata$logprice <- log(housedata$price)
housedata$logdriveshop <- log(housedata$driveshop)

#### Fit a covariate only model
form <- logprice~ns(crime,3)+rooms+sales+factor(type) + logdriveshop
model <- lm(formula=form, data=housedata)

#### Compute a Moran's I test on the residuals
W.nb <- poly2nb(spatialhousedata, row.names = rownames(housedata))
W.list <- nb2listw(W.nb, style="B")
resid.model <- residuals(model)
moran.mc(x=resid.model, listw=W.list, nsim=1000)

#### Run a regression model with spatially correlated random effects
W.mat <- nb2mat(W.nb, style="B")
## Not run:
model.spatial <- S.CARleroux(formula=form, data=housedata,
family="gaussian", W=W.mat, burnin=10000, n.sample=40000,
verbose=FALSE)
print(model.spatial)

## End(Not run)

```

combine.data.shapefile

Combine a data frame with a shapefile to create a SpatialPolygons-DataFrame object.

Description

This function combines a data frame with a shapefile and creates a `SpatialPolygonsDataFrame` object from the 'sp' package. The creation of this object allows the variables in the data frame to be plotted as a spatial map using the 'spplot()' function. For the function to work the row names

in the data frame must be a subset of the names in the first column of the dbf file. See the main CARBayes help file for an example.

Usage

```
combine.data.shapefile(data, shp, dbf)
```

Arguments

| | |
|------|--|
| data | A data frame containing the variables you wish to map. Note, the row names of this data frame must appear in the .dbf file (see argument 3 below). |
| shp | The .shp part of a shapefile containing the polygons of the areas that the data relate to. |
| dbf | The .dbf part of the shapefile containing a lookup table, whose first column includes the row names of the data frame. |

Value

SpatialPolygonsDataFrame

A SpatialPolygonsDataFrame object from the 'sp' package, which has combined the data frame with the shapefile and allows the variables in the data frame to be plotted as a spatial map using the `splot()` function.

Author(s)

Duncan Lee

Examples

```
## See the examples in the main help file for CARBayes.
```

| | |
|-------------------|---|
| highlight.borders | <i>Create a SpatialPoints object identifying a subset of borders between neighbouring areas, which allows them to be overlaid on a spatial map.</i> |
|-------------------|---|

Description

Create a SpatialPoints object identifying a subset of borders between neighbouring areas, which allows them to be overlaid on a spatial map. The borders that are identified could relate to neighbouring areas that exhibit very different values. See the main CARBayes help file for an example.

Usage

```
highlight.borders(border.locations, ID, shp, dbf)
```

Arguments

| | |
|-------------------------------|---|
| <code>border.locations</code> | An n by n matrix where n is the number of areas. The matrix contains 3 distinct values: NA for non-neighbouring areas, 0 for borders between neighbouring areas that are to be highlighted on the map, and 1 for borders between neighbouring areas that are not to be highlighted. |
| <code>ID</code> | A vector containing the row names of the data frame which are unique identifiers for each area. |
| <code>shp</code> | The .shp part of a shapefile containing the polygons of the areas that the data relate to. |
| <code>dbf</code> | The .dbf part of the shapefile containing a lookup table, whose first column includes the row names of the data frame. |

Value

| | |
|----------------------------|--|
| <code>SpatialPoints</code> | A <code>SpatialPoints</code> object from the 'sp' package, which contains the vertices of all the borders to be highlighted on the map. The mapping can be done using the <code>splot()</code> function, see the main help file in <code>CARBayes</code> for an example. |
|----------------------------|--|

Author(s)

Duncan Lee

Examples

```
## See the examples in the main help file for CARBayes.
```

```
print.carbayes          Print a summary of the fitted model to the screen.
```

Description

This function takes a `carbayes` object and returns a summary of the fitted model.

Usage

```
## S3 method for class 'carbayes'
print(x, ...)
```

Arguments

| | |
|------------------|---|
| <code>x</code> | A <code>carbayes</code> fitted model object |
| <code>...</code> | Ignored |

Author(s)

Duncan Lee

Examples

```
## See the examples in the main help file for CARBayes.
```

| | |
|----------|--|
| S.CARbym | <i>Fit a generalised linear model with a set of spatially autocorrelated random effect following the BYM conditional autoregressive (CAR) prior to spatial data.</i> |
|----------|--|

Description

The function fits a Bayesian hierarchical model with spatially correlated random effects to the data, where the data likelihood can be binomial, Gaussian or Poisson. The random effects are modelled by the BYM conditional autoregressive (CAR) model (Besag et. al. 1991). The model represents the linear predictor of the data by a combination of covariates and two sets of random effects. For the latter, the first set are independent, while the second are spatially correlated and come from the IAR model. A set of offsets can also be included on the linear predictor scale. Inference is based on Markov chain Monte Carlo (MCMC) simulation, using a combination of Gibbs sampling and Metropolis steps.

Usage

```
S.CARbym(formula, family, data=NULL, trials=NULL, W, burnin=0, n.sample=1000,
thin=1, prior.mean.beta=NULL, prior.var.beta=NULL, prior.nu2=NULL,
prior.tau2=NULL, prior.sigma2=NULL, verbose=TRUE)
```

Arguments

| | |
|----------|--|
| formula | A formula for the covariate part of the model, using the same notation as for the <code>lm()</code> function. The offsets should also be included here using the <code>offset()</code> function. |
| family | One of either 'binomial', 'gaussian' or 'poisson', which respectively specify a binomial likelihood model with a logistic link function, a Gaussian likelihood model with an identity link function, or a Poisson likelihood model with a log link function. |
| data | A data.frame containing the variables in the formula. |
| trials | Only used if family='binomial'. A vector the same length as the response containing the total number of trials for each area. |
| W | A binary n by n neighbourhood matrix (where n is the number of spatial units). The jkth element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. |
| burnin | The number of MCMC samples to discard as the burnin period. Defaults to 0. |
| n.sample | The number of MCMC samples to generate. Defaults to 1,000. |
| thin | The level of thinning to apply to the MCMC samples to reduce their temporal autocorrelation. Defaults to 1. |

| | |
|------------------------------|--|
| <code>prior.mean.beta</code> | A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros. |
| <code>prior.var.beta</code> | A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 1000. |
| <code>prior.nu2</code> | Only used for the Gaussian model. The prior shape and scale in the form of <code>c(shape, scale)</code> for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to <code>c(0.001, 0.001)</code> . |
| <code>prior.tau2</code> | The prior shape and scale in the form of <code>c(shape, scale)</code> for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to <code>c(0.001, 0.001)</code> . |
| <code>prior.sigma2</code> | The prior shape and scale in the form of <code>c(shape, scale)</code> for an Inverse-Gamma(shape, scale) prior for sigma2. Defaults to <code>c(0.001, 0.001)</code> . |
| <code>verbose</code> | Logical, should the function update the user on its progress. |

Details

For further details about how to apply the function see the examples below and in the vignette.

Value

| | |
|----------------------------------|---|
| <code>summary.results</code> | A summary table of the parameters. |
| <code>samples</code> | A list containing the MCMC samples from the model. |
| <code>fitted.values</code> | A vector of fitted values for each area. |
| <code>residuals</code> | A vector of residuals for each area. |
| <code>modelfit</code> | Model fit criteria including the Deviance Information Criterion (DIC), the effective number of parameters in the model(p.d), and the Log Marginal Predictive Likelihood (LMPL). |
| <code>accept</code> | The acceptance probabilities for the parameters. |
| <code>localised.structure</code> | NULL, for compatability with the other models. |
| <code>formula</code> | The formula for the covariate and offset part of the model. |
| <code>model</code> | A text string describing the model fit. |
| <code>X</code> | The design matrix of covariates. |

Author(s)

Duncan Lee

References

Besag, J., J. York, and A. Mollie (1991). Bayesian image restoration with two applications in spatial statistics. *Annals of the Institute of Statistics and Mathematics* 43, 1-59.

Examples

```
#####
#### Run the model on simulated data on a lattice
#####

#### Set up a square lattice region
x.easting <- 1:10
x.northing <- 1:10
Grid <- expand.grid(x.easting, x.northing)
n <- nrow(Grid)

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <-array(0, c(n,n))
W <-array(0, c(n,n))
  for(i in 1:n)
  {
  for(j in 1:n)
  {
  temp <- (Grid[i,1] - Grid[j,1])^2 + (Grid[i,2] - Grid[j,2])^2
  distance[i,j] <- sqrt(temp)
  if(temp==1) W[i,j] <- 1
  }
  }

#### Generate the covariates and response data
x1 <- rnorm(n)
x2 <- rnorm(n)
theta <- rnorm(n, sd=0.05)
phi <- mvrnorm(n=1, mu=rep(0,n), Sigma=0.4 * exp(-0.1 * distance))
logit <- x1 + x2 + theta + phi
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(50,n)
Y <- rbinom(n=n, size=trials, prob=prob)

#### Run the BYM model
formula <- Y ~ x1 + x2
## Not run: model <- S.CARbym(formula=formula, family="binomial, trials=trials,
W=W, burnin=5000, n.sample=10000)
## End(Not run)
```

S.CARcluster

Fit a generalised linear model with a set of spatially autocorrelated random effects following a conditional autoregressive (CAR) prior and a piecewise constant jump component proposed by Lee and Sarran (2014) to spatial data.

Description

The function fits a Bayesian hierarchical model with spatially correlated random effects and a cluster component to the data, where the data likelihood can be binomial, Gaussian or Poisson. The random effects are modelled by the conditional autoregressive (CAR) model proposed by Leroux et. al. (1999). The model represents the linear predictor of the data by a cluster component and a set of random effects. The latter are spatially correlated and come from the Leroux CAR model. Inference is based on Markov chain Monte Carlo (MCMC) simulation, using a combination of Gibbs sampling and Metropolis steps.

Usage

```
S.CARcluster(formula, exposure=NULL, family, data=NULL, G, trials=NULL, W,
burnin=0, n.sample=1000, thin=1, prior.mean.beta=NULL, prior.var.beta=NULL,
prior.mean.alpha=NULL, prior.var.alpha=NULL, prior.nu2=NULL, prior.tau2=NULL,
prior.delta=NULL, verbose=TRUE)
```

Arguments

| | |
|----------|--|
| formula | A formula for the covariate part of the model, using the same notation as for the <code>lm()</code> function. The offsets should also be included here using the <code>offset()</code> function. |
| exposure | This argument allows a single covariate to have variation in its values within an areal unit if the data are Poisson counts. This argument is not required and is only applicable if <code>family="poisson"</code> . If such a covariate exists the argument is a list object, in which each element in the list corresponds to each areal unit. Each element in the list is a matrix where the first column is the vector of values of the covariate and the second column is the vector of weights for each value. See Lee and Sarran (2014) for more details. |
| family | One of either 'binomial', 'gaussian' or 'poisson', which respectively specify a binomial likelihood model with a logistic link function, a Gaussian likelihood model with an identity link function, or a Poisson likelihood model with a log link function. |
| data | A <code>data.frame</code> containing the variables in the formula. |
| G | The number of clusters to fit to the data. Must be an integer with minimum value <code>G=2</code> . |
| trials | Only used if <code>family='binomial'</code> . A vector the same length as the response containing the total number of trials for each area. |
| W | A binary <code>n</code> by <code>n</code> neighbourhood matrix (where <code>n</code> is the number of spatial units). The <code>jkth</code> element equals one if areas <code>(j, k)</code> are spatially close (e.g. share a common border) and is zero otherwise. |
| burnin | The number of MCMC samples to discard as the burnin period. Defaults to 0. |
| n.sample | The number of MCMC samples to generate. Defaults to 1,000. |
| thin | The level of thinning to apply to the MCMC samples to reduce their temporal autocorrelation. Defaults to 1. |

| | |
|-------------------------------|---|
| <code>prior.mean.beta</code> | A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros. |
| <code>prior.var.beta</code> | A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 1000. |
| <code>prior.mean.alpha</code> | The prior mean for the regression parameter alpha (Gaussian priors are assumed) multiplying the covariate with within area variation in values. Defaults to zero. |
| <code>prior.var.alpha</code> | The prior variance for the regression parameter alpha (Gaussian priors are assumed) multiplying the covariate with within area variation in values. Defaults to 1000. |
| <code>prior.nu2</code> | The prior shape and scale in the form of $c(\text{shape}, \text{scale})$ for an Inverse-Gamma(shape, scale) prior for nu2. Only for the Gaussian model. Defaults to $c(0.001, 0.001)$. |
| <code>prior.tau2</code> | The prior shape and scale in the form of $c(\text{shape}, \text{scale})$ for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to $c(0.001, 0.001)$. |
| <code>prior.delta</code> | The prior maximum for the cluster smoothing parameter delta. Defaults to 10. |
| <code>verbose</code> | Logical, should the function update the user on its progress. |

Details

For further details about how to apply the function see the examples below and in the vignette.

Value

| | |
|----------------------------------|---|
| <code>summary.results</code> | A summary table of the parameters. |
| <code>samples</code> | A list containing the MCMC samples from the model. |
| <code>fitted.values</code> | A vector of fitted values for each area. |
| <code>residuals</code> | A vector of residuals for each area. |
| <code>modelfit</code> | Model fit criteria including the Deviance Information Criterion (DIC), the effective number of parameters in the model(p.d), and the Log Marginal Predictive Likelihood (LMPL). |
| <code>accept</code> | The acceptance probabilities for the parameters. |
| <code>localised.structure</code> | A vector giving the posterior median of which cluster group each area is in. |
| <code>formula</code> | The formula for the covariate and offset part of the model. |
| <code>model</code> | A text string describing the model fit. |
| <code>X</code> | The design matrix of covariates. |

Author(s)

Duncan Lee

References

Leroux, B., X. Lei, and N. Breslow (1999). Estimation of disease rates in small areas: A new mixed model for spatial dependence, Chapter Statistical Models in Epidemiology, the Environment and Clinical Trials, Halloran, M and Berry, D (eds), pp. 135-178. Springer-Verlag, New York.

Lee, D and Sarran, C (2014). Controlling for unmeasured confounding and spatial misalignment in long-term air pollution and health studies. arXiv:1412.4479.

Examples

```
#####
#### Run the model on simulated data - localised CAR model
#####

#### Set up a square lattice region
x.easting <- 1:10
x.northing <- 1:10
Grid <- expand.grid(x.easting, x.northing)
n <- nrow(Grid)

#### Split the area into two groups between which there will be a boundary.
groups <- rep(1, n)
groups[Grid$Var1>5] <- 2

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <- array(0, c(n,n))
W <- array(0, c(n,n))
for(i in 1:n)
  {
for(j in 1:n)
  {
temp <- (Grid[i,1] - Grid[j,1])^2 + (Grid[i,2] - Grid[j,2])^2
distance[i,j] <- sqrt(temp)
if(temp==1) W[i,j] <- 1
}
}

#### Generate the response data
phi <- mvrnorm(n=1, mu=groups, Sigma=0.2 * exp(-0.1 * distance))
logit <- phi
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(50,n)
Y <- rbinom(n=n, size=trials, prob=prob)

#### Run the localised smoothing model
formula <- Y ~ 1
## Not run: model <- S.CARcluster(formula=formula, family="poisson", G=2,
W=W,burnin=5000, n.sample=10000)
## End(Not run)
```

S.CARdissimilarity *Fit a generalised linear model with a set of spatially autocorrelated random effect following the localised conditional autoregressive (CAR) prior proposed by Lee and Mitchell (2012) to spatial data.*

Description

The function fits a Bayesian hierarchical model with spatially correlated random effects to the data, where the data likelihood can be binomial, Gaussian or Poisson. The random effects are modelled by a localised conditional autoregressive (CAR) model, which is a generalisation of that proposed by Lee and Mitchell (2012). The random effects in neighbouring areas (e.g. those that share a common border) are modelled as correlated or conditionally independent, depending on whether the populations living in the two areas are similar (correlated random effects) or very different (conditionally independent). The model represents the linear predictor by a combination of covariates and a set of random effects. Inference is based on Markov chain Monte Carlo (MCMC) simulation, using a combination of Gibbs sampling and Metropolis steps.

Usage

```
S.CARdissimilarity(formula, family, data=NULL, trials=NULL, W,
Z, burnin=0, n.sample=1000, thin=1, prior.mean.beta=NULL,
prior.var.beta=NULL, prior.nu2=NULL, prior.tau2=NULL, verbose=TRUE)
```

Arguments

| | |
|----------|--|
| formula | A formula for the covariate part of the model, using the same notation as for the <code>lm()</code> function. The offsets should also be included here using the <code>offset()</code> function. |
| family | One of either 'binomial', 'gaussian' or 'poisson', which respectively specify a binomial likelihood model with a logistic link function, a Gaussian likelihood model with an identity link function, or a Poisson likelihood model with a log link function. |
| data | A <code>data.frame</code> containing the variables in the formula. |
| trials | A vector the same length as the response containing the total number of trials for each area. |
| W | A binary n by n neighbourhood matrix (where n is the number of spatial units). The jk th element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. |
| Z | A list, where each element is an n by n matrix of non-negative dissimilarity metrics. For details of their construction see the main help page for <code>CARBayes</code> . |
| burnin | The number of MCMC samples to discard as the burnin period. Defaults to 0. |
| n.sample | The number of MCMC samples to generate. Defaults to 1,000. |
| thin | The level of thinning to apply to the MCMC samples to reduce their temporal autocorrelation. Defaults to 1. |

| | |
|------------------------------|--|
| <code>prior.mean.beta</code> | A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros. |
| <code>prior.var.beta</code> | A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 1000. |
| <code>prior.nu2</code> | Only used for the Gaussian model. The prior shape and scale in the form of <code>c(shape, scale)</code> for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to <code>c(0.001, 0.001)</code> . |
| <code>prior.tau2</code> | The prior shape and scale in the form of <code>c(shape, scale)</code> for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to <code>c(0.001, 0.001)</code> . |
| <code>verbose</code> | Logical, should the function update the user on its progress. |

Details

For further details about how to apply the function see the examples below and in the vignette.

Value

| | |
|----------------------------------|--|
| <code>summary.results</code> | A summary table of the parameters. |
| <code>samples</code> | A list containing the MCMC samples from the model. |
| <code>fitted.values</code> | A vector of fitted values for each area. |
| <code>residuals</code> | A vector of residuals for each area. |
| <code>modelfit</code> | Model fit criteria including the Deviance Information Criterion (DIC), the effective number of parameters in the model(p.d), and the Log Marginal Predictive Likelihood (LMPL). |
| <code>accept</code> | The acceptance probabilities for the parameters. |
| <code>localised.structure</code> | A list containing two matrices: <code>W.posterior</code> contains posterior medians for each element <code>w_{kj}</code> of the <code>n</code> by <code>n</code> neighbourhood matrix <code>W</code> ; <code>W.border.prob</code> contains posterior probabilities that each <code>w_{kj}</code> element of the <code>n</code> by <code>n</code> neighbourhood matrix <code>W</code> equals zero. This corresponds to the posterior probability of a boundary in the random effects surface. In both cases elements which correspond to two non-neighbouring areas have NA values. |
| <code>formula</code> | The formula for the covariate and offset part of the model. |
| <code>model</code> | A text string describing the model fit. |
| <code>X</code> | The design matrix of covariates. |

Author(s)

Duncan Lee

References

Lee, D. and R. Mitchell (2012). Boundary detection in disease mapping studies. *Biostatistics*, 13, 415-426.

Examples

```
#####
#### Run the model on simulated data - localised CAR model
#####

#### Set up a square lattice region
x.easting <- 1:10
x.northing <- 1:10
Grid <- expand.grid(x.easting, x.northing)
n <- nrow(Grid)

#### Split the area into two groups between which there will be a boundary.
groups <- rep(1, n)
groups[Grid$Var1>5] <- 2

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <- array(0, c(n,n))
W <- array(0, c(n,n))
for(i in 1:n)
{
for(j in 1:n)
{
temp <- (Grid[i,1] - Grid[j,1])^2 + (Grid[i,2] - Grid[j,2])^2
distance[i,j] <- sqrt(temp)
if(temp==1) W[i,j] <- 1
}
}

#### Generate the response data
phi <- mvrnorm(n=1, mu=groups, Sigma=0.2 * exp(-0.1 * distance))
logit <- phi
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(50,n)
Y <- rbinom(n=n, size=trials, prob=prob)

#### Generate a dissimilarity metric
dissimilarity <- cbind(groups) + rnorm(n, sd=0.2)
dissimilarity.matrix <- as.matrix(dist(cbind(dissimilarity, dissimilarity),
method="manhattan", diag=TRUE, upper=TRUE)) * W/2

Z <- list(dissimilarity.matrix=dissimilarity.matrix)

#### Run the localised smoothing model
formula <- Y ~ 1
## Not run: model <- S.CARdissimilarity(formula=formula, family="binomial",
trials=trials, W=W, Z=Z, burnin=5000, n.sample=10000)
## End(Not run)
```

| | |
|----------|--|
| S.CARiar | <i>Fit a generalised linear model with a set of spatially autocorrelated random effect following the intrinsic conditional autoregressive (CAR) prior to spatial data.</i> |
|----------|--|

Description

The function fits a Bayesian hierarchical model with spatially correlated random effects to the data, where the data likelihood can be binomial, Gaussian or Poisson. The random effects are modelled by the intrinsic conditional autoregressive (IAR) model (Besag et. al. 1991). The model represents the linear predictor by a combination of covariates and a sets of random effects. The latter are spatially correlated and come from the intrinsic CAR model. A set of offsets can also be included on the linear predictor scale. Inference is based on Markov chain Monte Carlo (McMC) simulation, using a combination of Gibbs sampling and Metropolis steps.

Usage

```
S.CARiar(formula, family, data=NULL, trials=NULL, W, burnin=0, n.sample=1000,
thin=1, prior.mean.beta=NULL, prior.var.beta=NULL, prior.nu2=NULL,
prior.tau2=NULL, verbose=TRUE)
```

Arguments

| | |
|-----------------|--|
| formula | A formula for the covariate part of the model, using the same notation as for the <code>lm()</code> function. The offsets should also be included here using the <code>offset()</code> function. |
| family | One of either 'binomial', 'gaussian' or 'poisson', which respectively specify a binomial likelihood model with a logistic link function, a Gaussian likelihood model with an identity link function, or a Poisson likelihood model with a log link function. |
| data | A <code>data.frame</code> containing the variables in the formula. |
| trials | A vector the same length as the response containing the total number of trials for each area. |
| W | A binary n by n neighbourhood matrix (where n is the number of spatial units). The jk th element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. |
| burnin | The number of MCMC samples to discard as the burnin period. Defaults to 0. |
| n.sample | The number of MCMC samples to generate. Defaults to 1,000. |
| thin | The level of thinning to apply to the MCMC samples to reduce their temporal autocorrelation. Defaults to 1. |
| prior.mean.beta | A vector of prior means for the regression parameters β (Gaussian priors are assumed). Defaults to a vector of zeros. |
| prior.var.beta | A vector of prior variances for the regression parameters β (Gaussian priors are assumed). Defaults to a vector with values 1000. |

| | |
|------------|--|
| prior.nu2 | Only used for the Gaussian model. The prior shape and scale in the form of $c(\text{shape}, \text{scale})$ for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to $c(0.001, 0.001)$. |
| prior.tau2 | The prior shape and scale in the form of $c(\text{shape}, \text{scale})$ for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to $c(0.001, 0.001)$. |
| verbose | Logical, should the function update the user on its progress. |

Details

For further details about how to apply the function see the examples below and in the vignette.

Value

| | |
|---------------------|---|
| summary.results | A summary table of the parameters. |
| samples | A list containing the MCMC samples from the model. |
| fitted.values | A vector of fitted values for each area. |
| residuals | A vector of residuals for each area. |
| modelfit | Model fit criteria including the Deviance Information Criterion (DIC), the effective number of parameters in the model(p.d), and the Log Marginal Predictive Likelihood (LMPL). |
| accept | The acceptance probabilities for the parameters. |
| localised.structure | NULL, for compatability with the other models. |
| formula | The formula for the covariate and offset part of the model. |
| model | A text string describing the model fit. |
| X | The design matrix of covariates. |

Author(s)

Duncan Lee

References

Besag, J., J. York, and A. Mollie (1991). Bayesian image restoration with two applications in spatial statistics. *Annals of the Institute of Statistics and Mathematics* 43, 1-59.

Examples

```
#####
#### Run the model on simulated data on a lattice
#####

#### Set up a square lattice region
x.easting <- 1:10
x.northing <- 1:10
Grid <- expand.grid(x.easting, x.northing)
```

```

n <- nrow(Grid)

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <-array(0, c(n,n))
W <-array(0, c(n,n))
for(i in 1:n)
{
  for(j in 1:n)
  {
    temp <- (Grid[i,1] - Grid[j,1])^2 + (Grid[i,2] - Grid[j,2])^2
    distance[i,j] <- sqrt(temp)
    if(temp==1) W[i,j] <- 1
  }
}

#### Generate the covariates and response data
x1 <- rnorm(n)
x2 <- rnorm(n)
theta <- rnorm(n, sd=0.05)
phi <- mvrnorm(n=1, mu=rep(0,n), Sigma=0.4 * exp(-0.1 * distance))
logit <- x1 + x2 + theta + phi
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(50,n)
Y <- rbinom(n=n, size=trials, prob=prob)

#### Run the IAR model
formula <- Y ~ x1 + x2
## Not run: model <- S.CARlar(formula=formula, family="binomial", trials=trials,
W=W, burnin=5000, n.sample=10000)
## End(Not run)

```

S.CARleroux

Fit a generalised linear model with a set of spatially autocorrelated random effect following the conditional autoregressive (CAR) prior proposed by Leroux et al. (1999) to spatial data.

Description

The function fits a Bayesian hierarchical model with spatially correlated random effects to the data, where the data likelihood can be binomial, Gaussian or Poisson. The random effects are modelled by the conditional autoregressive (CAR) model proposed by Leroux et. al. 1999. The model represents the linear predictor by a combination of covariates and a sets of random effects. The latter are spatially correlated and come from the Leroux CAR model. A set of offsets can also be included on the linear predictor scale. Inference is based on Markov chain Monte Carlo (McMC) simulation, using a combination of Gibbs sampling and Metropolis steps.

Usage

```
S.CARleroux(formula, family, data=NULL, trials=NULL, W, burnin=0,
n.sample=1000, thin=1, prior.mean.beta=NULL, prior.var.beta=NULL,
prior.nu2=NULL, prior.tau2=NULL, verbose=TRUE)
```

Arguments

| | |
|-----------------|--|
| formula | A formula for the covariate part of the model, using the same notation as for the <code>lm()</code> function. The offsets should also be included here using the <code>offset()</code> function. |
| family | One of either 'binomial', 'gaussian' or 'poisson', which respectively specify a binomial likelihood model with a logistic link function, a Gaussian likelihood model with an identity link function, or a Poisson likelihood model with a log link function. |
| data | A <code>data.frame</code> containing the variables in the formula. |
| trials | A vector the same length as the response containing the total number of trials for each area. |
| W | A binary n by n neighbourhood matrix (where n is the number of spatial units). The jk th element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. |
| burnin | The number of MCMC samples to discard as the burnin period. Defaults to 0. |
| n.sample | The number of MCMC samples to generate. Defaults to 1,000. |
| thin | The level of thinning to apply to the MCMC samples to reduce their temporal autocorrelation. Defaults to 1. |
| prior.mean.beta | A vector of prior means for the regression parameters β (Gaussian priors are assumed). Defaults to a vector of zeros. |
| prior.var.beta | A vector of prior variances for the regression parameters β (Gaussian priors are assumed). Defaults to a vector with values 1000. |
| prior.nu2 | Only used for the Gaussian model. The prior shape and scale in the form of $c(\text{shape}, \text{scale})$ for an Inverse-Gamma(shape, scale) prior for ν_2 . Defaults to $c(0.001, 0.001)$. |
| prior.tau2 | The prior shape and scale in the form of $c(\text{shape}, \text{scale})$ for an Inverse-Gamma(shape, scale) prior for τ_2 . Defaults to $c(0.001, 0.001)$. |
| verbose | Logical, should the function update the user on its progress. |

Details

For further details about how to apply the function see the examples below and in the vignette.

Value

| | |
|-----------------|--|
| summary.results | A summary table of the parameters. |
| samples | A list containing the MCMC samples from the model. |

fitted.values A vector of fitted values for each area.
 residuals A vector of residuals for each area.
 modelfit Model fit criteria including the Deviance Information Criterion (DIC), the effective number of parameters in the model(p.d), and the Log Marginal Predictive Likelihood (LMPL).
 accept The acceptance probabilities for the parameters.
 localised.structure NULL, for compatability with the other models.
 formula The formula for the covariate and offset part of the model.
 model A text string describing the model fit.
 X The design matrix of covariates.

Author(s)

Duncan Lee

References

Leroux, B., X. Lei, and N. Breslow (1999). Estimation of disease rates in small areas: A new mixed model for spatial dependence, Chapter Statistical Models in Epidemiology, the Environment and Clinical Trials, Halloran, M and Berry, D (eds), pp. 135-178. Springer-Verlag, New York.

Examples

```
#####
#### Run the model on simulated data on a lattice
#####

#### Set up a square lattice region
x.easting <- 1:10
x.northing <- 1:10
Grid <- expand.grid(x.easting, x.northing)
n <- nrow(Grid)

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <-array(0, c(n,n))
W <-array(0, c(n,n))
for(i in 1:n)
{
  for(j in 1:n)
  {
    temp <- (Grid[i,1] - Grid[j,1])^2 + (Grid[i,2] - Grid[j,2])^2
    distance[i,j] <- sqrt(temp)
    if(temp==1) W[i,j] <- 1
  }
}

#### Generate the covariates and response data
```

```

x1 <- rnorm(n)
x2 <- rnorm(n)
theta <- rnorm(n, sd=0.05)
phi <- mvrnorm(n=1, mu=rep(0,n), Sigma=0.4 * exp(-0.1 * distance))
logit <- x1 + x2 + theta + phi
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(50,n)
Y <- rbinom(n=n, size=trials, prob=prob)

#### Run the Leroux model
formula <- Y ~ x1 + x2
## Not run: model <- S.CARleroux(formula=formula, family="binomial",
trials=trials, W=W, burnin=5000, n.sample=10000)
## End(Not run)

```

| | |
|---------------|--|
| S.independent | <i>Fit a generalised linear model with a set of independent random effect to spatial data.</i> |
|---------------|--|

Description

The function fits a Bayesian hierarchical model with independent random effects to the data, where the data likelihood can be binomial, Gaussian or Poisson. The model represents the linear predictor by a combination of covariates and a set of independent random effects. A set of offsets can also be included on the linear predictor scale. Inference is based on Markov chain Monte Carlo (MCMC) simulation, using a combination of Gibbs sampling and Metropolis steps.

Usage

```

S.independent(formula, family, data=NULL, trials=NULL, burnin=0,
n.sample=1000, thin=1, prior.mean.beta=NULL, prior.var.beta=NULL,
prior.nu2=NULL, prior.sigma2=NULL, verbose=TRUE)

```

Arguments

| | |
|---------|--|
| formula | A formula for the covariate part of the model, using the same notation as for the <code>lm()</code> function. The offsets should also be included here using the <code>offset()</code> function. |
| family | One of either 'binomial', 'gaussian' or 'poisson', which respectively specify a binomial likelihood model with a logistic link function, a Gaussian likelihood model with an identity link function, or a Poisson likelihood model with a log link function. |
| data | A data.frame containing the variables in the formula. |
| trials | A vector the same length as the response containing the total number of trials for each area. |
| burnin | The number of MCMC samples to discard as the burnin period. Defaults to 0. |

| | |
|------------------------------|--|
| <code>n.sample</code> | The number of MCMC samples to generate. Defaults to 1,000. |
| <code>thin</code> | The level of thinning to apply to the MCMC samples to reduce their temporal autocorrelation. Defaults to 1. |
| <code>prior.mean.beta</code> | A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros. |
| <code>prior.var.beta</code> | A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 1000. |
| <code>prior.nu2</code> | Only used for the Gaussian model. The prior shape and scale in the form of <code>c(shape, scale)</code> for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to <code>c(0.001, 0.001)</code> . |
| <code>prior.sigma2</code> | The prior shape and scale in the form of <code>c(shape, scale)</code> for an Inverse-Gamma(shape, scale) prior for sigma2. Defaults to <code>c(0.001, 0.001)</code> . |
| <code>verbose</code> | Logical, should the function update the user on its progress. |

Details

For further details about how to apply the function see the examples below and in the vignette.

Value

| | |
|----------------------------------|---|
| <code>summary.results</code> | A summary table of the parameters. |
| <code>samples</code> | A list containing the MCMC samples from the model. |
| <code>fitted.values</code> | A vector of fitted values for each area. |
| <code>residuals</code> | A vector of residuals for each area. |
| <code>modelfit</code> | Model fit criteria including the Deviance Information Criterion (DIC), the effective number of parameters in the model(p.d), and the Log Marginal Predictive Likelihood (LMPL). |
| <code>accept</code> | The acceptance probabilities for the parameters. |
| <code>localised.structure</code> | NULL, for compatibility with the other models. |
| <code>formula</code> | The formula for the covariate and offset part of the model. |
| <code>model</code> | A text string describing the model fit. |
| <code>X</code> | The design matrix of covariates. |

Author(s)

Duncan Lee

Examples

```
#####
#### Run the model on simulated data on a lattice
#####

#### Set up a square lattice region
x.easting <- 1:10
x.northing <- 1:10
Grid <- expand.grid(x.easting, x.northing)
n <- nrow(Grid)

#### Generate the covariates and response data
x1 <- rnorm(n)
x2 <- rnorm(n)
theta <- rnorm(n, sd=0.05)
logit <- x1 + x2 + theta
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(50,n)
Y <- rbinom(n=n, size=trials, prob=prob)

#### Run the independent model
formula <- Y ~ x1 + x2
## Not run: model <- S.independent(formula=formula, family="binomial", trials=trials,
burnin=5000, n.sample=10000)
## End(Not run)
```

| | |
|------------------|--|
| spatialhousedata | <i>A SpatialPolygonsDataFrame object (from the sp package) containing property price data for the 271 Intermediate Geographies (IG) in the Greater Glasgow and Clyde health board.</i> |
|------------------|--|

Description

A SpatialPolygonsDataFrame object containing data on median property prices in 2008 and possible covariates for the 270 Intermediate Geographies (IG) in the Greater Glasgow and Clyde health board. These data are used in the worked example in the main help page of CARBayes.

Usage

```
data(spatialhousedata)
```

Format

A SpatialPolygonsDataFrame object where the dataframe contains 271 observations on the following 6 variables.

price Median property price in each IG.

crime The crime rate (number of crimes per 10,000 people) in each IG.
rooms The median number of rooms in a property in each IG.
sales The percentage of properties that sold in each IG in a year.
driveshop The average time taken to drive to a shopping centre in minutes.
type The predominant property type in each IG with levels: detached, flat, semi, terrace

Source

These data were provided by the Scottish Government, and can also be downloaded from <http://www.sns.gov.uk/>. Further details about these data are also available at the same location.

Examples

```
data(spatialhousedata)
```

| | |
|-----------------|---|
| spatialrespdata | <i>A SpatialPolygonsDataFrame object (from the sp package) containing respiratory hospital admissions data for the 134 Intermediate Geographies (IG) to the north of the river Clyde in the Greater Glasgow and Clyde health board.</i> |
|-----------------|---|

Description

A SpatialPolygonsDataFrame object containing data on respiratory disease data for the 134 Intermediate Geographies (IG) to the north of the river Clyde in the Greater Glasgow and Clyde health board. These data are used in the CARBayes vignette.

Usage

```
data(spatialrespdata)
```

Format

A SpatialPolygonsDataFrame object where the dataframe contains 134 observations on the following 3 variables.

observed2010 The observed numbers of cases of hospitalisation due to respiratory disease in each IG in 2010.
expected2010 The expected numbers of cases of hospitalisation due to respiratory disease in each IG in 2010 computed using external standardisation from Scotland wide disease rates.
incomedep2010 The percentage of people in each IG who are defined to be income deprived.

Source

These data were provided by the Scottish Government, and can also be downloaded from <http://www.sns.gov.uk/>. Further details about these data are also available at the same location.

Examples

```
data(spatialrespdata)
```

| | |
|-------------------|---|
| summarise.lincomb | <i>Compute the posterior distribution and quantiles of a linear combination of the covariate component of the linear predictor.</i> |
|-------------------|---|

Description

This function takes in a CARBayes model object and computes the posterior distribution and quantiles of a linear combination of the covariate component of the linear predictor. For example, if in the model the user specified a quadratic effect of a covariate on the response, then this function allows you to compute the posterior distribution of the entire quadratic relationship.

Usage

```
summarise.lincomb(model, columns=NULL, quantiles=0.5, distribution=FALSE)
```

Arguments

| | |
|--------------|--|
| model | A CARBayes model object from fitting one of the models in this package. |
| columns | A vector of numbers stating which columns in the design matrix of covariates you wish to compute the posterior distribution for. |
| quantiles | A vector of posterior quantiles required for each column of MCMC samples. |
| distribution | A logical value stating whether the entire posterior distribution should be returned or just the specified quantiles. |

Details

For further details about how to apply the function see the example below.

Value

| | |
|-----------|---|
| quantiles | A 2 dimensional array containing the required posterior quantiles. Each row relates to a data value, and each column to a different requested quantile. |
| posterior | A 2 dimensional array containing the required posterior distribution. Each column relates to a different data value. Is NULL if distribution=FALSE in the function arguments. |

Author(s)

Duncan Lee

Examples

```
## See Vignette for an example of its use.
```

| | |
|-------------------|--|
| summarise.samples | <i>Summarise a matrix of Markov chain Monte Carlo samples.</i> |
|-------------------|--|

Description

This function takes in a matrix of Markov chain Monte Carlo (McMC) samples from a CARBayes model object, such as from a set of parameters or fitted values, and calculates posterior quantiles and exceedence probabilities. The latter are probabilities of the form $P(\text{quantity} > \text{cldata})$, where c is a threshold chosen by the user.

Usage

```
summarise.samples(samples, columns=NULL, quantiles=0.5, exceedences=NULL)
```

Arguments

| | |
|-------------|--|
| samples | A matrix of McMC samples resulting from fitting one of the models. The object must be of class <code>mcmc</code> from the <code>coda</code> package. |
| columns | A vector of numbers stating which columns in the matrix of McMC samples you wish to summarise. Defaults to all columns. |
| quantiles | A vector of posterior quantiles required for each column of McMC samples. |
| exceedences | A vector of threshold levels, c , that you wish to calculate exceedence probabilities for, for each column of McMC samples. |

Details

For further details about how to apply the function see the example below.

Value

| | |
|-------------|--|
| quantiles | A 2 dimensional array containing the required posterior quantiles. Each row relates to a parameter, and each column to a different requested quantile. If the argument <code>quantiles</code> is missing this object is <code>NULL</code> . |
| exceedences | A 2 dimensional array containing the required exceedence probabilities. Each row relates to a parameter, and each column to a different requested exceedence probability. If the argument <code>exceedences</code> is missing this object is <code>NULL</code> . |

Author(s)

Duncan Lee

Examples

```
## See Vignette for an example of its use.
```

Index

*Topic **datasets**

spatialhousedata, [23](#)

spatialrespdata, [24](#)

CARBayes (CARBayes-package), [2](#)

CARBayes-package, [2](#)

combine.data.shapefile, [4](#)

highlight.borders, [5](#)

print.carbayes, [6](#)

S.CARbym, [7](#)

S.CARcluster, [9](#)

S.CARDissimilarity, [13](#)

S.CARiar, [16](#)

S.CARleroux, [18](#)

S.independent, [21](#)

spatialhousedata, [23](#)

spatialrespdata, [24](#)

summarise.lincomb, [25](#)

summarise.samples, [26](#)