Package ‘CARBayes’

March 8, 2024

Type Package

Title Spatial Generalised Linear Mixed Models for Areal Unit Data

Version 6.1.1

Date 2024-03-08

Author Duncan Lee

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

Description Implements a class of univariate and multivariate spatial generalised linear mixed models for areal unit data, with inference in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation using a single or multiple Markov chains. The response variable can be binomial, Gaussian, multinomial, Poisson or zero-inflated Poisson (ZIP), and spatial autocorrelation is modelled by a set of random effects that are assigned a conditional autoregressive (CAR) prior distribution. A number of different models are available for univariate spatial data, including models with no random effects as well as random effects modelled by different types of CAR prior, including the BYM model (Besag et al., 1991, <doi:10.1007/BF00116466>) and Leroux model (Leroux et al., 2000, <doi:10.1007/978-1-4612-1284-3_4>). Additionally, a multivariate CAR (MCAR) model for multivariate spatial data is available, as is a two-level hierarchical model for modelling data relating to individuals within areas. Full details are given in the 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 has been supported by the Engineering and Physical Science Research Council (EPSRC) grant EP/J017442/1, ESRC grant ES/K006460/1, Innovate UK / Natural Environment Research Council (NERC) grant NE/N007352/1 and the TB Alliance.

License GPL (>= 2)

Depends MASS, R (>= 3.5.0), Rcpp (>= 0.11.5)

Imports CARBayesdata, coda, dplyr, GGally, glmnet, igraph, mapview, MCMCpack, parallel, RColorBrewer, sf, spam, spdep, stats, truncnorm, utils

LinkingTo Rcpp

LazyLoad yes

ByteCompile yes

URL https://github.com/duncanplee/CARBayes

BugReports https://github.com/duncanplee/CARBayes/issues

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-03-08 13:20:02 UTC

R topics documented:

`CARBayes-package` *Spatial Generalised Linear Mixed Models for Areal Unit Data*

Description

Implements a class of univariate and multivariate spatial generalised linear mixed models for areal unit data, with inference in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation using a single or multiple Markov chains. The response variable can be binomial, Gaussian, multinomial, Poisson or zero-inflated Poisson (ZIP), and spatial autocorrelation is modelled by a set of random effects that are assigned a conditional autoregressive (CAR) prior distribution. A number of different models are available for univariate spatial data, including models with no random effects as well as random effects modelled by different types of CAR prior, including the BYM model (Besag et al., 1991, <doi:10.1007/BF00116466>) and the Leroux model (Leroux et al., 2000, <doi:10.1007/978-1-4612-1284-3_4>). Additionally, a multivariate CAR (MCAR) model for multivariate spatial data is available, as is a two-level hierarchical model for modelling data relating to individuals within areas. Full details are given in the 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 has been supported by the Engineering and Physical Science Research Council (EPSRC) grant EP/J017442/1, ESRC grant ES/K006460/1, Innovate UK / Natural Environment Research Council (NERC) grant NE/N007352/1 and the TB Alliance.

Details

Package:	CARBayes
Type:	Package
Version:	6.1.1
Date:	2024-03-08
License:	GPL (>= 2)

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.

Gelfand, A and Vounatsou, P (2003). Proper multivariate conditional autoregressive models for spatial data analysis, Biostatistics, 4, 11-25.

Kavanagh, L., D. Lee, and G. Pryce (2016). Is Poverty Decentralising? Quantifying Uncertainty in the Decentralisation of Urban Poverty, Annals of the American Association of Geographers, 106, 1286-1298.

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

Lee, D and Sarran, C (2015). Controlling for unmeasured confounding and spatial misalignment in long-term air pollution and health studies, Environmetrics, 26, 477-487.

Lee, D (2024). Computationally efficient localised spatial smoothing of disease rates using anisotropic basis functions and penalised regression fitting, Spatial Statistics, 59, 100796.

Leroux B, Lei X, Breslow N (2000). "Estimation of Disease Rates in SmallAreas: A New Mixed Model for Spatial Dependence." In M Halloran, D Berry (eds.), *Statistical Models in Epidemiology, the Environment and Clinical Trials*, pp. 179-191. Springer-Verlag, New York.

Roberts, G and Rosenthal, J (1998). Optimal scaling of discrete approximations to the Langevin diffusions, Journal of the Royal Statistical Society Series B 60, 255-268.

Examples

```
## See the examples in the function specific help files and in the vignette
## accompanying this package.
```

`fitted.CARBayes` *Extract the fitted values from a model.*

Description

This function takes a CARBayes object and returns the vector of fitted values (posterior means).

Usage

```
    ## S3 method for class 'CARBayes'
fitted(object, ...)
```

Arguments

`object` A CARBayes fitted model object.

`...` Ignored.

Author(s)

Duncan Lee

`highlight.borders` *Creates an sf data.frame object (from the sf package) identifying a subset of borders between neighbouring areas.*

Description

Creates an sf data.frame object (from the sf package) identifying a subset of borders between neighbouring areas, which allows them to be overlayed on a map. An example is given in the vignette accompanying this package.

Usage

```
highlight.borders(border.locations, sfdata)
```

Arguments

`border.locations`
A K by K matrix, where K is the number of areas, containing 3 distinct values: NA for non-neighbouring areas; 0 for borders between neighbouring areas to be highlighted on a map; and 1 for borders between neighbouring areas not to be highlighted on a map.

`sfdata` An sf data.frame (from the sf package) object used for plotting the data and creating the original neighbourhood matrix W.

Value

An sf data.frame object from the sf package, which contains the vertices of all the borders to be highlighted on the map. The mapping can be done using the mapview package, see the vignette accompanying this package for an example.

Author(s)

Duncan Lee

Examples

```
## See the vignette accompanying this package for an example of its use.
```

logLik.CARBayes *Extract the estimated loglikelihood from a fitted model.*

Description

This function takes a CARBayes object and returns the estimated loglikelihood (posterior means).

Usage

```
    ## S3 method for class 'CARBayes'
logLik(object, ...)
```

Arguments

`object`	A CARBayes fitted model object.
`...`	Ignored.

Author(s)

Duncan Lee

model.matrix.CARBayes *Extract the model (design) matrix from a model.*

Description

This function takes a CARBayes object and returns the design matrix.

Usage

```
    ## S3 method for class 'CARBayes'
model.matrix(object, ...)
```

Arguments

object	A CARBayes fitted model object.
...	Ignored.

Author(s)

Duncan Lee

MVS.CARleroux *Fit a multivariate spatial generalised linear mixed model to data, where the random effects are modelled by a multivariate conditional autoregressive model.*

Description

Fit a multivariate spatial generalised linear mixed model to areal unit data, where the response variable can be binomial, Gaussian, multinomial or Poisson. The linear predictor is modelled by known covariates and a vector of random effects. The latter account for both spatial and between variable correlation, via a Kronecker product formulation. Spatial correlation is captured by the conditional autoregressive (CAR) prior proposed by Leroux et al. (2000), and between variable correlation is captured by a between variable covariance matrix with no fixed structure. This is a type of multivariate conditional autoregressive (MCAR) model. Further details are given in the vignette accompanying this package. Independent (over space) random effects can be obtained by setting rho=0, while the intrinsic MCAR model can be obtained by setting rho=1. Inference is conducted in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. Missing (NA) values are allowed in the response, and posterior predictive distributions are created for the missing values using data augmentation. These are saved in the "samples" argument in the output of the function and are denoted by "Y". For the multinomial model the first category in the multinomial data (first column of the response matrix) is taken as the baseline, and the covariates are linearly related to the log of the ratio (theta_j / theta_1) for j=1,...,J, where theta_j is the probability of being in category j. For a full model specification see the vignette accompanying this package.

Usage

```
MVS.CARleroux(formula, family, data=NULL,  trials=NULL, W, burnin, n.sample,
thin=1, n.chains=1, n.cores=1, prior.mean.beta=NULL, prior.var.beta=NULL,
prior.nu2=NULL, prior.Sigma.df=NULL, prior.Sigma.scale=NULL, rho=NULL, MALA=TRUE,
verbose=TRUE)
```

Arguments

`formula` A formula for the covariate part of the model using the syntax of the lm() function. Offsets can be included here using the offset() function. The response and the offset (if included) should be matrices of dimension K*J, where K is the number of spatial units and J is the number of different variables (categories in the multinomial model). The covariates should each be a K*1 vector, and different regression parameters are estimated for each of the J variables. Missing (NA) values are allowed in the response.

`family` One of either "binomial", "gaussian", "multinomial", or "poisson", which respectively specify a binomial likelihood model with a logistic link function, a Gaussian likelihood model with an identity link function, a multinomial likelihood model with a logistic link function, or a Poisson likelihood model with a log link function.

`data` An optional data.frame containing the variables in the formula.

`trials` Only used if family="binomial" or family="multinomial". For the binomial family it is a K*J matrix matrix the same dimension as the response. A the multinomial family it is a vector of length K.

`W` A non-negative K by K neighbourhood matrix (where K is the number of spatial units). Typically a binary specification is used, where the jkth element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. The matrix can be non-binary, but each row must contain at least one non-zero entry.

`burnin` The number of MCMC samples to discard as the burn-in period in each chain.

`n.sample` The overall number of MCMC samples to generate in each chain.

`thin` The level of thinning to apply to the MCMC samples in each chain to reduce their autocorrelation. Defaults to 1 (no thinning).

`n.chains` The number of MCMC chains to run when fitting the model. Defaults to 1.

`n.cores` The number of computer cores to run the MCMC chains on. Must be less than or equal to n.chains. Defaults to 1.

`prior.mean.beta`
A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros.

`prior.var.beta` A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 100,000.

`prior.nu2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to c(1, 0.01) and only used if family="Gaussian".

`prior.Sigma.df` The degrees of freedom for the Inverse-Wishart prior formulation for the covariance matrix Sigma. This prior formulation follows the marginally weakly-informative specification proposed by Huang and Wand (2013). Defaults to 2, which corresponds to non-informative uniform priors on the interval [-1,1] being assigned to each correlation parameter within the Sigma matrix.

`prior.Sigma.scale`
The J times 1 vector of prior scales for the square roots of the diagonal elements of the covariance matrix Sigma. This prior formulation is the marginally weakly-informative prior specification proposed by Huang and Wand (2013). Thus, the jth element of this vector is the scale parameter for the zero centred half-t prior (with shape given by prior.Sigma.df) assumed for the standard deviation of the random effects corresponding to the jth outcome. Defaults to a vector of values of 100,000.

`rho` The value in the interval [0, 1] that the spatial dependence parameter rho is fixed at if it should not be estimated. If this arugment is NULL then rho is estimated in the model.

`MALA` Logical, should the function use Metropolis adjusted Langevin algorithm (MALA) updates (TRUE, default) or simple random walk updates (FALSE) for the regression parameters. Not applicable if family="gaussian" or family="multinomial".

`verbose` Logical, should the function update the user on its progress.

Value

`summary.results`
A summary table of the parameters.

`samples` A list containing the MCMC samples from the model.

`fitted.values` A matrix of fitted values based on posterior means from the model. Each row of a matrix relates to an area and each column to a response (category).

`residuals` A list with 2 elements, where each element is a matrix of a type of residuals. Each row of a matrix relates to an area and each column to a response (category). The types of residual are "response" (raw), and "pearson".

`modelfit` Model fit criteria including the Deviance Information Criterion (DIC) and its corresponding estimated effective number of parameters (p.d), the Log Marginal Predictive Likelihood (LMPL), the Watanabe-Akaike Information Criterion (WAIC) and its corresponding estimated number of effective parameters (p.w), and the loglikelihood.

`accept` The acceptance probabilities for the parameters.

`localised.structure`
NULL, for compatability with other models.

`formula` The formula (as a text string) for the response, covariate and offset parts of the model

`model` A text string describing the model fit.

`mcmc.info` A vector giving details of the numbers of MCMC samples generated.

`X` The design matrix of covariates.

Author(s)

Duncan Lee

References

Gelfand, A and Vounatsou, P (2003). Proper multivariate conditional autoregressive models for spatial data analysis, Biostatistics, 4, 11-25.

Huang, A., and Wand, M (2013). Simple Marginally Noninformative Prior Distributions for Covariance Matrices. Bayesian Analysis, 8, 439-452.

Kavanagh, L., D. Lee, and G. Pryce (2016). Is Poverty Decentralising? Quantifying Uncertainty in the Decentralisation of Urban Poverty, Annals of the American Association of Geographers, 106, 1286-1298.

Leroux B, Lei X, Breslow N (2000). "Estimation of Disease Rates in SmallAreas: A New Mixed Model for Spatial Dependence." In M Halloran, D Berry (eds.), *Statistical Models in Epidemiology, the Environment and Clinical Trials*, pp. 179-191. Springer-Verlag, New York.

Examples

```
#################################################
#### Run the model on simulated data on a lattice
#################################################
#### Load other libraries required
library(MASS)

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

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <- as.matrix(dist(Grid))
W <-array(0, c(K,K))
W[distance==1] <-1
K <- nrow(W)

#### Generate the correlation structures
Q.W <- 0.99 * (diag(apply(W, 2, sum)) - W) + 0.01 * diag(rep(1,K))
Q.W.inv <- solve(Q.W)

Sigma <- matrix(c(1,0.5,0, 0.5,1,0.3, 0, 0.3, 1), nrow=3)
Sigma.inv <- solve(Sigma)
J <- nrow(Sigma)
N.all <- K * J

precision.phi <- kronecker(Q.W, Sigma.inv)
var.phi <- solve(precision.phi)

#### Generate the covariate component
x1 <- rnorm(K)
```

```
x2 <- rnorm(K)
XB <- cbind(0.1 * x1 - 0.1*x2, -0.1 * x1 + 0.1*x2, 0.1 * x1 - 0.1*x2)

#### Generate the random effects
phi <- mvrnorm(n=1, mu=rep(0,N.all), Sigma=var.phi)


#### Generate the response data
lp <-as.numeric(t(XB)) + phi
prob <- exp(lp) / (1 + exp(lp))
trials.vec <- rep(100,N.all)
Y.vec <- rbinom(n=N.all, size=trials.vec, prob=prob)

#### Turn the data and trials into matrices where each row is an area.
Y <- matrix(Y.vec, nrow=K, ncol=J, byrow=TRUE)
trials <- matrix(trials.vec, nrow=K, ncol=J, byrow=TRUE)

#### Run the Leroux model
formula <- Y ~ x1 + x2
## Not run: model <- MVS.CARleroux(formula=formula, family="binomial",
trials=trials, W=W, burnin=20000, n.sample=100000)
## End(Not run)

#### Toy example for checking
model <- MVS.CARleroux(formula=formula, family="binomial",
trials=trials, W=W, burnin=10, n.sample=50)
```

print.CARBayes *Print a summary of a fitted CARBayes model to the screen.*

Description

This function takes a CARBayes object and returns a summary of the fitted model. The summary includes, for selected parameters, posterior means and 95 percent credible intervals, the effective number of independent samples and the Geweke convergence diagnostic in the form of a Z-score.

Usage

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

Arguments

x	A CARBayes fitted model object.
...	Ignored.s

Author(s)

Duncan Lee

`residuals.CARBayes` *Extract the residuals from a model.*

Description

This function takes a CARBayes object and returns a set of residuals. The allowable types of residual are "response" (raw) and "pearson" (the default). In each case the fitted values are based on posterior means.

Usage

```
    ## S3 method for class 'CARBayes'
residuals(object, type, ...)
```

Arguments

`object`	A CARBayes fitted model object.
`type`	A text string and one of "response" or "pearson". If this argument is omitted the default is "pearson".
`...`	Ignored.

Author(s)

Duncan Lee

`S.CARbym` *Fit a spatial generalised linear mixed model to data, where the random effects have a BYM conditional autoregressive prior.*

Description

Fit a spatial generalised linear mixed model to areal unit data, where the response variable can be binomial, Poisson, or zero-inflated Poisson (ZIP). Note, a Gaussian likelihood is not allowed because of a lack of identifiability among the parameters. The linear predictor is modelled by known covariates and 2 vectors of random effects. The latter are modelled by the BYM conditional autoregressive prior proposed by Besag et al. (1991), and further details are given in the vignette accompanying this package. Inference is conducted in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. Missing (NA) values are allowed in the response, and posterior predictive distributions are created for the missing values using data augmentation. These are saved in the "samples" argument in the output of the function and are denoted by "Y". For the ZIP model covariates can be used to estimate the probability of an observation being a structural zero, via a logistic regression equation. For a full model specification see the vignette accompanying this package.

Usage

```
S.CARbym(formula, formula.omega=NULL, family, data=NULL, trials=NULL, W, burnin,
n.sample, thin=1, n.chains=1, n.cores=1, prior.mean.beta=NULL,
prior.var.beta=NULL, prior.tau2=NULL, prior.sigma2=NULL, prior.mean.delta=NULL,
prior.var.delta=NULL, MALA=TRUE, verbose=TRUE)
```

Arguments

`formula` A formula for the covariate part of the model using the syntax of the lm() function. Offsets can be included here using the offset() function. The response, offset and each covariate are vectors of length K*1. The response can contain missing (NA) values.

`formula.omega` A one-sided formula object with no response variable (left side of the "~") needed, specifying the covariates in the logistic regression model for modelling the probability of an observation being a structural zero. Each covariate (or an offset) needs to be a vector of length K*1. Only required for zero-inflated Poisson models.

`family` One of either "binomial","poisson" or "zip", which respectively specify a binomial likelihood model with a logistic link function, a Poisson likelihood model with a log link function, or a zero-inflated Poisson model with a log link function.

`data` An optional 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. Only used if family="binomial".

`W` A non-negative K by K neighbourhood matrix (where K is the number of spatial units). Typically a binary specification is used, where the jkth element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. The matrix can be non-binary, but each row must contain at least one non-zero entry.

`burnin` The number of MCMC samples to discard as the burn-in period in each chain.

`n.sample` The overall number of MCMC samples to generate in each chain.

`thin` The level of thinning to apply to the MCMC samples in each chain to reduce their autocorrelation. Defaults to 1 (no thinning).

`n.chains` The number of MCMC chains to run when fitting the model. Defaults to 1.

`n.cores` The number of computer cores to run the MCMC chains on. Must be less than or equal to n.chains. Defaults to 1.

`prior.mean.beta`
A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros.

`prior.var.beta` A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 100,000.

`prior.tau2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to c(1, 0.01).

`prior.sigma2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for sigma2. Defaults to c(1, 0.01).

`prior.mean.delta`
A vector of prior means for the regression parameters delta (Gaussian priors are assumed) for the zero probability logistic regression component of the model. Defaults to a vector of zeros. Only used if family="multinomial".

`prior.var.delta`
A vector of prior variances for the regression parameters delta (Gaussian priors are assumed) for the zero probability logistic regression component of the model. Defaults to a vector with values 100,000. Only used if family="multinomial".

`MALA` Logical, should the function use Metropolis adjusted Langevin algorithm (MALA) updates (TRUE, default) or simple random walk updates (FALSE) for the regression parameters.

`verbose` Logical, should the function update the user on its progress.

Value

`summary.results`
A summary table of the parameters.

`samples` A list containing the MCMC samples from the model.

`fitted.values` The fitted values based on posterior means from the model.

`residuals` A matrix with 2 columns where each column is a type of residual and each row relates to an area. The types are "response" (raw), and "pearson".

`modelfit` Model fit criteria including the Deviance Information Criterion (DIC) and its corresponding estimated effective number of parameters (p.d), the Log Marginal Predictive Likelihood (LMPL), the Watanabe-Akaike Information Criterion (WAIC) and its corresponding estimated number of effective parameters (p.w), and the loglikelihood.

`accept` The acceptance probabilities for the parameters.

`localised.structure`
NULL, for compatability with other models.

`formula` The formula (as a text string) for the response, covariate and offset parts of the model

`model` A text string describing the model fit.

`mcmc.info` A vector giving details of the numbers of MCMC samples generated.

`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
#################################################
#### Load other libraries required
library(MASS)

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

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <- as.matrix(dist(Grid))
W <-array(0, c(K,K))
W[distance==1] <-1

#### Generate the covariates and response data
x1 <- rnorm(K)
x2 <- rnorm(K)
theta <- rnorm(K, sd=0.05)
phi <- mvrnorm(n=1, mu=rep(0,K), Sigma=0.4 * exp(-0.1 * distance))
logit <- x1 + x2 + theta + phi
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(50,K)
Y <- rbinom(n=K, 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=20000, n.sample=100000)
## End(Not run)

#### Toy example for checking
model <- S.CARbym(formula=formula, family=”binomial”, trials=trials,
W=W, burnin=20, n.sample=50)
```

`S.CARdissimilarity` *Fit a spatial generalised linear mixed model to data, where the random effects have a localised conditional autoregressive prior.*

Description

Fit a spatial generalised linear mixed model to areal unit data, where the response variable can be binomial, Gaussian or Poisson. The linear predictor is modelled by known covariates and a vector of random effects. The latter are modelled by the localised conditional autoregressive prior proposed by Lee and Mitchell (2012), and further details are given in the vignette accompanying this package.

Inference is conducted in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. Missing (NA) values are allowed in the response, and posterior predictive distributions are created for the missing values using data augmentation. These are saved in the "samples" argument in the output of the function and are denoted by "Y". For a full model specification see the vignette accompanying this package.

Usage

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

Arguments

`formula` A formula for the covariate part of the model using the syntax of the lm() function. Offsets can be included here using the offset() function. The response, offset and each covariate is a vector of length K*1. The response can contain missing (NA) values.

`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` An optional 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. Only used if family="binomial".

`W` A non-negative K by K neighbourhood matrix (where K is the number of spatial units). Typically a binary specification is used, where the jkth element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. For this model only the matrix must be binary.

`Z` A list, where each element is a K by K matrix of non-negative dissimilarity metrics.

`W.binary` Logical, should the estimated neighbourhood matrix have only binary (0,1) values.

`burnin` The number of MCMC samples to discard as the burn-in period in each chain.

`n.sample` The overall number of MCMC samples to generate in each chain.

`thin` The level of thinning to apply to the MCMC samples in each chain to reduce their autocorrelation. Defaults to 1 (no thinning).

`n.chains` The number of MCMC chains to run when fitting the model. Defaults to 1.

`n.cores` The number of computer cores to run the MCMC chains on. Must be less than or equal to n.chains. Defaults to 1.

`prior.mean.beta` A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros.

`prior.var.beta` A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 100,000.

`prior.nu2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to c(1, 0.01) and only used if family="Gaussian".

`prior.tau2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to c(1, 0.01).

`MALA` Logical, should the function use Metropolis adjusted Langevin algorithm (MALA) updates (TRUE, default) or simple random walk updates (FALSE) for the regression parameters. Not applicable if family="gaussian".

`verbose` Logical, should the function update the user on its progress.

Value

`summary.results` A summary table of the parameters.

`samples` A list containing the MCMC samples from the model.

`fitted.values` The fitted values based on posterior means from the model.

`residuals` A matrix with 2 columns where each column is a type of residual and each row relates to an area. The types are "response" (raw), and "pearson".

`modelfit` Model fit criteria including the Deviance Information Criterion (DIC) and its corresponding estimated effective number of parameters (p.d), the Log Marginal Predictive Likelihood (LMPL), the Watanabe-Akaike Information Criterion (WAIC) and its corresponding estimated number of effective parameters (p.w), and the loglikelihood.

`accept` The acceptance probabilities for the parameters.

`localised.structure` A list containing two matrices: W.posterior contains posterior medians for each element w_kj of the K by K neighbourhood matrix W; W.border.prob contains posterior probabilities that each w_kj element of the K by K neighbourhood matrix W equals zero. This corresponds to the posterior probability of a boundary in the random effects surface. The latter is only present if W.binary=TRUE, otherwise it is missing (NA). In all cases W elements that correspond to two non-neighbouring areas have NA values.

`formula` The formula (as a text string) for the response, covariate and offset parts of the model

`model` A text string describing the model fit.

`mcmc.info` A vector giving details of the numbers of MCMC samples generated.

`X` 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 on a lattice
##################################################
#### Load other libraries required
library(MASS)

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

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

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <- as.matrix(dist(Grid))
W <-array(0, c(K,K))
W[distance==1] <-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,K)
Y <- rbinom(n=K, size=trials, prob=prob)


#### Generate a dissimilarity metric
dissimilarity <- cbind(groups) + rnorm(K, 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, W.binary=TRUE, burnin=20000, n.sample=100000)
## End(Not run)

#### Toy example for checking
model <- S.CARdissimilarity(formula=formula, family="binomial",
trials=trials, W=W, Z=Z, W.binary=TRUE, burnin=10, n.sample=50)
```

S.CARleroux *Fit a spatial generalised linear mixed model to data, where the random effects have a Leroux conditional autoregressive prior.*

Description

Fit a spatial generalised linear mixed model to areal unit data, where the response variable can be binomial, Gaussian, Poisson or zero-inflated Poisson (ZIP). The linear predictor is modelled by known covariates and a vector of random effects. The latter are modelled by the conditional autoregressive prior proposed by Leroux et al. (2000), and further details are given in the vignette accompanying this package. Independent random effects can be obtained by setting rho=0, while the intrinsic CAR model can be obtained by setting rho=1. Inference is conducted in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. Missing (NA) values are allowed in the response, and posterior predictive distributions are created for the missing values using data augmentation. These are saved in the"samples" argument in the output of the function and are denoted by "Y". For the ZIP model covariates can be used to estimate the probability of an observation being a structural zero, via a logistic regression equation. For a full model specification see the vignette accompanying this package.

Usage

```
S.CARleroux(formula, formula.omega=NULL, family, data=NULL,  trials=NULL, W,
burnin, n.sample, thin=1, n.chains=1,  n.cores=1, prior.mean.beta=NULL,
prior.var.beta=NULL, prior.nu2=NULL, prior.tau2=NULL, prior.mean.delta=NULL,
prior.var.delta=NULL, rho=NULL, MALA=TRUE, verbose=TRUE)
```

Arguments

`formula` A formula for the covariate part of the model using the syntax of the lm() function. Offsets can be included here using the offset() function. The response, offset and each covariate is a vector of length K*1. The response can contain missing (NA) values.

`formula.omega` A one-sided formula object with no response variable (left side of the "~") needed, specifying the covariates in the logistic regression model for modelling the probability of an observation being a structural zero. Each covariate (or an offset) needs to be a vector of length K*1. Only required for zero-inflated Poisson models.

`family` One of either "binomial", "gaussian", "poisson" or "zip", which respectively specify a binomial likelihood model with a logistic link function, a Gaussian likelihood model with an identity link function, a Poisson likelihood model with a log link function, or a zero-inflated Poisson model with a log link function.

`data` An optional 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. Only used if family="binomial".

`W` A non-negative K by K neighbourhood matrix (where K is the number of spatial units). Typically a binary specification is used, where the jkth element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. The matrix can be non-binary, but each row must contain at least one non-zero entry.

`burnin` The number of MCMC samples to discard as the burn-in period in each chain.

`n.sample` The overall number of MCMC samples to generate in each chain.

`thin` The level of thinning to apply to the MCMC samples in each chain to reduce their autocorrelation. Defaults to 1 (no thinning).

`n.chains` The number of MCMC chains to run when fitting the model. Defaults to 1.

`n.cores` The number of computer cores to run the MCMC chains on. Must be less than or equal to n.chains. Defaults to 1.

`prior.mean.beta` A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros.

`prior.var.beta` A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 100,000.

`prior.nu2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to c(1, 0.01) and only used if family="Gaussian".

`prior.tau2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to c(1, 0.01).

`prior.mean.delta` A vector of prior means for the regression parameters delta (Gaussian priors are assumed) for the zero probability logistic regression component of the model. Defaults to a vector of zeros. Only used if family="multinomial".

`prior.var.delta` A vector of prior variances for the regression parameters delta (Gaussian priors are assumed) for the zero probability logistic regression component of the model. Defaults to a vector with values 100,000. Only used if family="multinomial".

`rho` The value in the interval [0, 1] that the spatial dependence parameter rho is fixed at if it should not be estimated. If this arugment is NULL then rho is estimated in the model.

`MALA` Logical, should the function use Metropolis adjusted Langevin algorithm (MALA) updates (TRUE, default) or simple random walk updates (FALSE) for the regression parameters. Not applicable if family="gaussian".

`verbose` Logical, should the function update the user on its progress.

Value

`summary.results` A summary table of the parameters.

`samples` A list containing the MCMC samples from the model.

`fitted.values` The fitted values based on posterior means from the model.

`residuals` A matrix with 2 columns where each column is a type of residual and each row relates to an area. The types are "response" (raw), and "pearson".

`modelfit` Model fit criteria including the Deviance Information Criterion (DIC) and its corresponding estimated effective number of parameters (p.d), the Log Marginal Predictive Likelihood (LMPL), the Watanabe-Akaike Information Criterion (WAIC) and its corresponding estimated number of effective parameters (p.w), and the loglikelihood.

`accept` The acceptance probabilities for the parameters.

`localised.structure`
NULL, for compatability with other models.

`formula` The formula (as a text string) for the response, covariate and offset parts of the model. If family="zip" this also includes the zero probability logistic regression formula.

`model` A text string describing the model fit.

`mcmc.info` A vector giving details of the numbers of MCMC samples generated.

`X` The design matrix of covariates.

Author(s)

Duncan Lee

References

Leroux B, Lei X, Breslow N (2000). "Estimation of Disease Rates in SmallAreas: A New Mixed Model for Spatial Dependence." In M Halloran, D Berry (eds.), *Statistical Models in Epidemiology, the Environment and Clinical Trials*, pp. 179-191. Springer-Verlag, New York.

Examples

```
#################################################
#### Run the model on simulated data on a lattice
#################################################
#### Load other libraries required
library(MASS)

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

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <- as.matrix(dist(Grid))
W <-array(0, c(K,K))
W[distance==1] <-1

#### Generate the covariates and response data
x1 <- rnorm(K)
x2 <- rnorm(K)
theta <- rnorm(K, sd=0.05)
phi <- mvrnorm(n=1, mu=rep(0,K), Sigma=0.4 * exp(-0.1 * distance))
logit <- x1 + x2 + theta + phi
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(50,K)
Y <- rbinom(n=K, 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=20000, n.sample=100000)
## End(Not run)

#### Toy example for checking
model <- S.CARleroux(formula=formula, family="binomial",
trials=trials, W=W, burnin=10, n.sample=50)
```

`S.CARlocalised` *Fit a spatial generalised linear mixed model to data, where a set of spatially smooth random effects are augmented with a piecewise constant intercept process.*

Description

Fit a spatial generalised linear mixed model to areal unit data, where the response variable can be binomial or Poisson. Note, a Gaussian likelihood is not allowed because of a lack of identifiability among the parameters. The linear predictor is modelled by known covariates, a vector of random effects and a piecewise constant intercept process. The random effects are modelled by an intrinsic CAR prior, while the piecewise constant intercept process was proposed by Lee and Sarran (2015), and allow neighbouring areas to have very different values. Further details are given in the vignette accompanying this package. Inference is conducted in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. Missing (NA) values are not allowed in this model. For a full model specification see the vignette accompanying this package.

Usage

```
S.CARlocalised(formula, family, data=NULL, G, trials=NULL, W, burnin, n.sample,
thin=1, n.chains=1, n.cores=1, prior.mean.beta=NULL, prior.var.beta=NULL,
prior.tau2=NULL,prior.delta=NULL, MALA=TRUE, verbose=TRUE)
```

Arguments

`formula` A formula for the covariate part of the model using the syntax of the lm() function. Offsets can be included here using the offset() function. The response, offset and each covariate is a vector of length K*1.The response cannot contain missing (NA) values.

`family` One of either "binomial" or "poisson", which respectively specify a binomial likelihood model with a logistic link function, or a Poisson likelihood model with a log link function.

`data` An optional data.frame containing the variables in the formula.

`G` The maximum number of distinct intercept terms (groups) to allow in the model.

`trials` A vector the same length as the response containing the total number of trials for each area. Only used if family="binomial".

`W` A non-negative K by K neighbourhood matrix (where K is the number of spatial units). Typically a binary specification is used, where the jkth element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. The matrix can be non-binary, but each row must contain at least one non-zero entry.

`burnin` The number of MCMC samples to discard as the burn-in period in each chain.

`n.sample` The overall number of MCMC samples to generate in each chain.

`thin` The level of thinning to apply to the MCMC samples in each chain to reduce their autocorrelation. Defaults to 1 (no thinning).

`n.chains` The number of MCMC chains to run when fitting the model. Defaults to 1.

`n.cores` The number of computer cores to run the MCMC chains on. Must be less than or equal to n.chains. Defaults to 1.

`prior.mean.beta` A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros.

`prior.var.beta` A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 100,000.

`prior.tau2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to c(1, 0.01).

`prior.delta` The prior maximum for the cluster smoothing parameter delta. Defaults to 10.

`MALA` Logical, should the function use Metropolis adjusted Langevin algorithm (MALA) updates (TRUE, default) or simple random walk updates (FALSE) for the regression parameters.

`verbose` Logical, should the function update the user on its progress.

Value

`summary.results` A summary table of the parameters.

`samples` A list containing the MCMC samples from the model.

`fitted.values` The fitted values based on posterior means from the model.

`residuals` A matrix with 2 columns where each column is a type of residual and each row relates to an area. The types are "response" (raw), and "pearson".

`modelfit` Model fit criteria including the Deviance Information Criterion (DIC) and its corresponding estimated effective number of parameters (p.d), the Log Marginal Predictive Likelihood (LMPL), the Watanabe-Akaike Information Criterion (WAIC) and its corresponding estimated number of effective parameters (p.w), and the loglikelihood.

`accept` The acceptance probabilities for the parameters.

`localised.structure` A vector giving the posterior median of which intercept group each area is in.

`formula` The formula (as a text string) for the response, covariate and offset parts of the model

`model` A text string describing the model fit.

`mcmc.info` A vector giving details of the numbers of MCMC samples generated.

`X` The design matrix of covariates.

Author(s)

Duncan Lee

References

Lee, D and Sarran, C (2015). Controlling for unmeasured confounding and spatial misalignment in long-term air pollution and health studies, Environmetrics, 26, 477-487.

Examples

```
##################################################
#### Run the model on simulated data on a lattice
##################################################
#### Load other libraries required
library(MASS)

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

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

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <- as.matrix(dist(Grid))
W <-array(0, c(K,K))
W[distance==1] <-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,K)
Y <- rbinom(n=K, size=trials, prob=prob)


#### Run the localised smoothing model
formula <- Y ~ 1
## Not run: model <- S.CARlocalised(formula=formula, family="binomial", trials=trials,
G=2, W=W,burnin=20000, n.sample=100000)
## End(Not run)

#### Toy example for checking
model <- S.CARlocalised(formula=formula, family="binomial", trials=trials,
G=2, W=W,burnin=10, n.sample=50)
```

S.CARmultilevel *Fit a spatial generalised linear mixed model to multi-level areal unit data, where the spatial random effects have a Leroux conditional autoregressive prior.*

Description

Fit a spatial generalised linear mixed model to multi-level areal unit data, where the response variable can be binomial, Gaussian or Poisson. The data are structured with individuals within areal units, and different numbers of individuals are allowed within each areal unit. The linear predictor is modelled by known covariates (either individual or areal level) and a vector of areal level random effects that are modelled by the conditional autoregressive prior proposed by Leroux et al. (2000). Independent random effects can be obtained by setting rho=0, while the intrinsic CAR model can be obtained by setting rho=1. Inference is conducted in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. Missing (NA) values are allowed in the response, and posterior predictive distributions are created for the missing values using data augmentation. These are saved in the "samples" argument in the output of the function and are denoted by "Y". For a full model specification see the vignette accompanying this package.

Usage

```
S.CARmultilevel(formula, family, data=NULL,  trials=NULL, W, ind.area,
burnin, n.sample, thin=1, n.chains=1, n.cores=1, prior.mean.beta=NULL,
prior.var.beta=NULL, prior.nu2=NULL, prior.tau2=NULL, rho=NULL, MALA=TRUE,
verbose=TRUE)
```

Arguments

`formula` A formula for the covariate part of the model using the syntax of the lm() function. Offsets can be included here using the offset() function. The response, offset and each covariate are vectors with length equal to the number of individuals. The response can contain missing (NA) values.

`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` An optional 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 individual. Only used if family="binomial".

`W` A non-negative K by K neighbourhood matrix (where K is the number of spatial units). Typically a binary specification is used, where the jkth element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. The matrix can be non-binary, but each row must contain at least one non-zero entry.

ind.area A vector of integers the same length as the number of data points (individuals) giving which spatial unit (nunmbered from 1 to K to align with the rows of the W matrix) each individual belongs to.

burnin The number of MCMC samples to discard as the burn-in period in each chain.

n.sample The overall number of MCMC samples to generate in each chain.

thin The level of thinning to apply to the MCMC samples in each chain to reduce their autocorrelation. Defaults to 1 (no thinning).

n.chains The number of MCMC chains to run when fitting the model. Defaults to 1.

n.cores The number of computer cores to run the MCMC chains on. Must be less than or equal to n.chains. Defaults to 1.

prior.mean.beta
A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros.

prior.var.beta A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 100,000.

prior.nu2 The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to c(1, 0.01) and only used if family="Gaussian".

prior.tau2 The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to c(1, 0.01).

rho The value in the interval [0, 1] that the spatial dependence parameter rho is fixed at if it should not be estimated. If this arugment is NULL then rho is estimated in the model.

MALA Logical, should the function use Metropolis adjusted Langevin algorithm (MALA) updates (TRUE, default) or simple random walk updates (FALSE) for the regression parameters. Not applicable if family="gaussian".

verbose Logical, should the function update the user on its progress.

Value

summary.results
A summary table of the parameters.

samples A list containing the MCMC samples from the model.

fitted.values The fitted values based on posterior means from the model.

residuals A matrix with 2 columns where each column is a type of residual and each row relates to an area. The types are "response" (raw), and "pearson".

modelfit Model fit criteria including the Deviance Information Criterion (DIC) and its corresponding estimated effective number of parameters (p.d), the Log Marginal Predictive Likelihood (LMPL), the Watanabe-Akaike Information Criterion (WAIC) and its corresponding estimated number of effective parameters (p.w), and the loglikelihood.

localised.structure
NULL, for compatability with other models.

formula The formula (as a text string) for the response, covariate and offset parts of the model

`model` A text string describing the model fit.

`mcmc.info` A vector giving details of the numbers of MCMC samples generated.

`X` 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)
K <- nrow(Grid)

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <- as.matrix(dist(Grid))
W <-array(0, c(K,K))
W[distance==1] <-1

#### Generate the number of individuals per area and which individuals to which areas
n <- sample(5:30, K, replace=TRUE)
n.total <- sum(n)
ind.area.temp <- rep(1:K, n)
ind.area <- sample(ind.area.temp, n.total, replace=FALSE)

#### Generate the covariates and response data
x1 <- rnorm(n.total)
x2 <- rnorm(n.total)
phi <- mvrnorm(n=1, mu=rep(0,K), Sigma=0.4 * exp(-0.1 * distance))
phi.extend <- phi[ind.area]
logit <- x1 + x2 + phi.extend
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(50,n.total)
Y <- rbinom(n=n.total, size=trials, prob=prob)


#### Run the model
formula <- Y ~ x1 + x2
## Not run: model <- S.CARmultilevel(formula=formula, family="binomial", ind.area=ind.area,
                trials=trials, W=W, burnin=20000, n.sample=100000)
## End(Not run)

#### Toy example for checking
model <- S.CARmultilevel(formula=formula, family="binomial", ind.area=ind.area,
                trials=trials, W=W, burnin=10, n.sample=50)
```

S.glm *Fit a generalised linear model to data.*

Description

Fit a generalised linear model to data, where the response variable can be binomial, Gaussian, multinomial, Poisson or zero-inflated Poisson (ZIP). Inference is conducted in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. Missing (NA) values are allowed in the response, and posterior predictive distributions are created for the missing values via data augmentation. These are saved in the "samples" argument in the output of the function and are denoted by "Y". For the multinomial model the first category in the multinomial data (first column of the response matrix) is taken as the baseline, and the covariates are linearly related to the log of the ratio (theta_j / theta_1) for j=1,...,J, where theta_j is the probability of being in category j. For the ZIP model covariates can be used to estimate the probability of an observation being a structural zero, via a logistic regression equation. For a full model specification see the vignette accompanying this package.

Usage

```
S.glm(formula, formula.omega=NULL, family, data=NULL,  trials=NULL, burnin,
n.sample, thin=1, n.chains=1, n.cores=1, prior.mean.beta=NULL,
prior.var.beta=NULL, prior.nu2=NULL, prior.mean.delta=NULL, prior.var.delta=NULL,
MALA=TRUE, verbose=TRUE)
```

Arguments

`formula` A formula for the covariate part of the model using the syntax of the lm() function. Offsets can be included here using the offset() function. The response, offset and each covariate are vectors of length K*1. For the multinomial model the response and the offset (if included) should be matrices of dimension K*J, where K is the number of spatial units and J is the number of different variables (categories in the multinomial model). The covariates should each be a K*1 vector, and different regression parameters are estimated for each of the J variables. The response can contain missing (NA) values.

`formula.omega` A one-sided formula object with no response variable (left side of the "~") needed, specifying the covariates in the logistic regression model for modelling the probability of an observation being a structural zero. Each covariate (or an offset) needs to be a vector of length K*1. Only required for zero-inflated Poisson models.

`family` One of either "binomial", "gaussian", "multinomial", "poisson" or "zip", which respectively specify a binomial likelihood model with a logistic link function, a Gaussian likelihood model with an identity link function, a multinomial likelihood model with a logistic link function, a Poisson likelihood model with a log link function, or a zero-inflated Poisson model with a log link function.

`data` An optional 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 data point. Only used if family="binomial" or family="multinomial".

`burnin` The number of MCMC samples to discard as the burn-in period in each chain.

`n.sample` The overall number of MCMC samples to generate in each chain.

`thin` The level of thinning to apply to the MCMC samples in each chain to reduce their autocorrelation. Defaults to 1 (no thinning).

`n.chains` The number of MCMC chains to run when fitting the model. Defaults to 1.

`n.cores` The number of computer cores to run the MCMC chains on. Must be less than or equal to n.chains. Defaults to 1.

`prior.mean.beta` A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros.

`prior.var.beta` A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 100,000.

`prior.nu2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to c(1, 0.01) and only used if family="Gaussian".

`prior.mean.delta` A vector of prior means for the regression parameters delta (Gaussian priors are assumed) for the zero probability logistic regression component of the model. Defaults to a vector of zeros. Only used if family="multinomial".

`prior.var.delta` A vector of prior variances for the regression parameters delta (Gaussian priors are assumed) for the zero probability logistic regression component of the model. Defaults to a vector with values 100,000. Only used if family="multinomial".

`MALA` Logical, should the function use Metropolis adjusted Langevin algorithm (MALA) updates (TRUE, default) or simple random walk updates (FALSE) for the regression parameters. Not applicable if family="gaussian" or family="multinomial".

`verbose` Logical, should the function update the user on its progress.

Value

`summary.results` A summary table of the parameters.

`samples` A list containing the MCMC samples from the model.

`fitted.values` The fitted values based on posterior means from the model. For the univariate data models this is a vector, while for the multivariate data models this is a matrix.

`residuals` If the family is "binomial", "gaussian" or "poisson", then this is a matrix with 2 columns, where each column is a type of residual and each row relates to an area. The types are "response" (raw), and "pearson". If family is "multinomial", then this is a list with 2 elements, where each element is a matrix of residuals of a different type. Each row of a matrix relates to an area and each column to a cateogry within the multinomial response. The types of residual are "response" (raw), and "pearson".

`modelfit` Model fit criteria including the Deviance Information Criterion (DIC) and its corresponding estimated effective number of parameters (p.d), the Log Marginal Predictive Likelihood (LMPL), the Watanabe-Akaike Information Criterion (WAIC) and its corresponding estimated number of effective parameters (p.w), and the loglikelihood.

`localised.structure`
NULL, for compatability with other models.

`formula` The formula (as a text string) for the response, covariate and offset parts of the model. If family="zip" this also includes the zero probability logistic regression formula.

`model` A text string describing the model fit.

`mcmc.info` A vector giving details of the numbers of MCMC samples generated.

`X` 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)
K <- nrow(Grid)

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

#### Run the model
formula <- Y ~ x1 + x2
## Not run: model <- S.glm(formula=formula, family=”binomial”, trials=trials,
burnin=20000, n.sample=100000)
## End(Not run)

#### Toy example for checking
model <- S.glm(formula=formula, family=”binomial”, trials=trials,
burnin=10, n.sample=50)
```

S.RAB *Fit a spatial generalised linear model with anisotropic basis functions to data for computationally efficient localised spatial smoothing, where the parameters are estimated by penalised maximum likelihood estimation with a ridge regression penalty.*

Description

Fit a spatial generalised linear model to areal unit data, where the response variable can be binomial, Gaussian or Poisson. The linear predictor is modelled by known covariates and a set of K anisotropic spatial basis functions. The basis functions are constructed from the set of geodesic distances between all pairs of areal units and a vector of ancillary data V, and the latter should have a similar spatial pattern to the residual (after covariate adjustment) spatial structure in the data on the linear predictor scale. Parameter estimtion is carried out via penalised maximum likelihood methods, and the basis function coefficients are constrained by a ridge regression penalty to prevent overfitting. The glmnet package is used for parameter estimation. Missing (NA) values are allowed in the response, and predictions are made for these values. This model implements localised spatial smoothing and allows for boundaries in the data surface using a computationally efficient approach.

Usage

```
S.RAB(formula, family, data=NULL, trials=NULL, W, V, nlambda=100, verbose=TRUE)
```

Arguments

`formula` A formula for the covariate part of the model using the syntax of the lm() function. Offsets can be included here using the offset() function. The response, offset and each covariate is a vector of length K*1. The response can contain missing (NA) values.

`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` An optional 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. Only used if family="binomial".

`W` A non-negative K by K neighbourhood matrix (where K is the number of spatial units). Typically a binary specification is used, where the jkth element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. The matrix can be non-binary, but each row must contain at least one non-zero entry.

`V` A vector of ancillary data of length K, which should have a similar spatial pattern to the residual (after covariate adjustment) spatial structure in the data on the linear predictor scale.

`nlambda` The number of possible values to use for the penalty parameter lambda in the glmnet() estimation function. Defaults to 100.

`verbose` Logical, should the function update the user on its progress.

Value

`beta.hat` The estimated regression parameters.

`sigma2.hat` The estimated error variance in the Gaussian data likelihood model. If a Gaussian model is not specified it is NA.

`lambda.hat` The estimated ridge regression penalty parameter.

`I` The level of residual spatial autocorrelation as measured by Moran's I statistic.

`fitted.values` The fitted values from the model.

`residuals` A matrix with 2 columns where each column is a type of residual and each row relates to an area. The types are "response" (raw), and "pearson".

`formula` The formula (as a text string) for the response, covariate and offset parts of the model.

`model.string` A text string describing the model fit.

`X` The design matrix of covariates and spatial basis functions.

`model` The fitted model object from the glmnet() function.

Author(s)

Duncan Lee

References

Lee, D (2024). Computationally efficient localised spatial smoothing of disease rates using anisotropic basis functions and penalised regression fitting, Spatial Statistics, 59, 100796.

Examples

```
#################################################
#### Run the model on simulated data on a lattice
#################################################
#### Load other libraries required
library(MASS)

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

#### set up distance and neighbourhood (W, based on sharing a common border) matrices
distance <- as.matrix(dist(Grid))
W <-array(0, c(K,K))
W[distance==1] <-1

#### Generate the spatial covariance structure
dists <- as.numeric(distance[upper.tri(distance)])
```

```
dists.quant <- quantile(dists, 0.05)
rho <- log(0.75) / -dists.quant
Sigma <- exp(-rho * distance)

#### Generate the boundaries
groups <-rep(0, K)
groups[Grid$Var2>5] <- 1

#### Generate the covariates and response data
x1 <- rnorm(K)
x2 <- rnorm(K)
phi <- mvrnorm(n=1, mu=rep(0,K), Sigma=0.1 * exp(-rho * distance))
logit <- x1 +  x2 + phi + 0.4 * groups
prob <- exp(logit) / (1 + exp(logit))
trials <- rep(100,K)
Y <- rbinom(n=K, size=trials, prob=prob)

#### Generate the ancillary data
V <- rnorm(n=K, mean=phi + 0.4*groups , sd=rep(0.05,K))

#### Run the RAB model
mod <- S.RAB(formula=Y~x1+x2, family="binomial", data=NULL, trials=trials, W=W,
        V=V, nlambda=50, verbose=TRUE)
```

Index