Package ‘RcppCensSpatial’

April 1, 2026

Type Package

Title Spatial Estimation and Prediction for Censored/Missing Responses

Version 1.0.0

Description It provides functions for estimating parameters in linear spatial models with censored or missing responses using the Expectation-Maximization (EM), Stochastic Approximation EM (SAEM), and Monte Carlo EM (MCEM) algorithms. These methods are widely used to obtain maximum likelihood (ML) estimates in the presence of incomplete data. The EM algorithm computes ML estimates when a closed-form expression for the conditional expectation of the complete-data log-likelihood is available. The MCEM algorithm replaces this expectation with a Monte Carlo approximation based on independent simulations of the missing data. In contrast, the SAEM algorithm decomposes the E-step into simulation and stochastic approximation steps, improving computational efficiency in complex settings. In addition, the package provides standard error estimation based on the Louis method. It also includes functionality for spatial prediction at new locations.
References used for this package: Galarza, C. E., Matos, L. A., Castro, L. M., & Lachos, V. H. (2022). Moments of the doubly truncated selection elliptical distributions with emphasis on the unified multivariate skew-t distribution. Journal of Multivariate Analysis, 189, 104944 <doi:10.1016/j.jmva.2021.104944>; Valeriano, K. A., Galarza, C. E., & Matos, L. A. (2023). Moments and random number generation for the truncated elliptical family of distributions. Statistics and Computing, 33(1), 32 <doi:10.1007/s11222-022-10200-4>.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

Imports ggplot2, gridExtra, MomTrunc, mvtnorm, Rcpp, Rdpack, relliptical, stats, StempCens

RdMacros Rdpack

LinkingTo RcppArmadillo, Rcpp, RcppProgress, roptim

Depends R (>= 2.10)

LazyData true

NeedsCompilation yes

Author Katherine A. L. Valeriano [aut, cre] (ORCID: <https://orcid.org/0000-0001-6388-4753>),
Christian Galarza Morales [ctb] (ORCID: <https://orcid.org/0000-0002-4818-6006>),
Larissa Avila Matos [ctb] (ORCID: <https://orcid.org/0000-0002-2635-0901>)

Maintainer Katherine A. L. Valeriano <katandreina@gmail.com>

Repository CRAN

Date/Publication 2026-03-31 22:40:42 UTC

Contents

`CovMat` *Covariance matrix for spatial models*

Description

It computes the spatial variance-covariance matrix using exponential, gaussian, matérn, or power exponential correlation function.

Usage

```
CovMat(phi, tau2, sig2, coords, type = "exponential", kappa = NULL)
```

Arguments

`phi`	spatial scaling parameter.
`tau2`	nugget effect parameter.
`sig2`	partial sill parameter.
`coords`	2D spatial coordinates of dimensions $n \times 2$.
`type`	type of spatial correlation function: `'exponential'`, `'gaussian'`, `'matern'`, and `'pow.exp'` for exponential, gaussian, matérn, and power exponential, respectively.
`kappa`	parameter for some spatial correlation functions. For exponential and gaussian `kappa=NULL`, for power exponential `0` < kappa <= `2`, and for matérn correlation function kappa > `0`.

Details

The spatial covariance matrix is given by

$\Sigma = [Cov(s_i, s_j)] = \sigma^2 R(\phi) + \tau^2 I_n$,

where $\sigma^2 > 0$ is the partial sill, $\phi > 0$ is the spatial scaling parameter, $\tau^2 > 0$ is known as the nugget effect in the geostatistical framework, $R(\phi)$ is the $n \times n$ correlation matrix computed from a correlation function, and I_n is the $n \times n$ identity matrix.

The spatial correlation functions available are:

Exponential: $Corr(d) = exp(-d/\phi)$,

Gaussian: $Corr(d) = exp(-(d/\phi)^2)$,

Matérn: $Corr(d) = \frac{1}{2^{(\kappa-1)}\Gamma(\kappa)}\left(\frac{d}{\phi}\right)^{\kappa} K_\kappa\left(\frac{d}{\phi}\right)$,

Power exponential: $Corr(d) = exp(-(d/\phi)^\kappa)$,

where $d \geq 0$ is the Euclidean distance between two observations, $\Gamma(.)$ is the gamma function, κ is the smoothness parameter, and $K_\kappa(.)$ is the modified Bessel function of the second kind of order κ.

Value

An $n \times n$ spatial covariance matrix.

Author(s)

Katherine L. Valeriano, Christian E. Galarza, and Larissa A. Matos.

See Also

dist2Dmatrix, EM.sclm, MCEM.sclm, SAEM.sclm

Examples

```
set.seed(1000)
n = 20
coords = round(matrix(runif(2*n, 0, 10), n, 2), 5)
Cov = CovMat(phi=5, tau2=0.8, sig2=2, coords=coords, type="exponential")
```

---

dist2Dmatrix *Distance matrix computation*

Description

It computes the Euclidean distance matrix for a set of coordinates.

Usage

```
dist2Dmatrix(coords)
```

Arguments

`coords` 2D spatial coordinates of dimensions $n \times 2$.

Value

An $n \times n$ distance matrix.

Author(s)

Katherine L. Valeriano, Christian E. Galarza, and Larissa A. Matos.

Examples

```
set.seed(1000)
n = 100
x = round(runif(n,0,10), 5)     # X coordinate
y = round(runif(n,0,10), 5)     # Y coordinate
Mdist = dist2Dmatrix(cbind(x, y))
```

`EM.sclm` *ML estimation of spatial censored linear models via the EM algorithm*

Description

It fits a spatial linear model with left-, right-, or interval-censored responses using the Expectation-Maximization (EM) algorithm. The function provides parameter estimates and their standard errors, and supports missing values in the response variable.

Usage

```
EM.sclm(y, x, ci, lcl = NULL, ucl = NULL, coords, phi0, nugget0,
  type = "exponential", kappa = NULL, lower = c(0.01, 0.01),
  upper = c(30, 30), MaxIter = 300, error = 1e-04, show_se = TRUE)
```

Arguments

- `y` vector of responses of length n.
- `x` design matrix of dimensions $n \times q$, where q is the number of fixed effects, including the intercept.
- `ci` vector of censoring indicators of length n. For each observation: `1` if censored/missing, `0` otherwise.
- `lcl, ucl` vectors of length n representing the lower and upper bounds of the interval, which contains the true value of the censored observation. Default `=NULL`, indicating no-censored data. For each observation: `lcl=-Inf` and `ucl=c` (left censoring); `lcl=c` and `ucl=Inf` (right censoring); and `lcl` and `ucl` must be finite for interval censoring. Moreover, missing data could be defined by setting `lcl=-Inf` and `ucl=Inf`.

`coords`	2D spatial coordinates of dimensions $n \times 2$.
`phi0`	initial value for the spatial scaling parameter.
`nugget0`	initial value for the nugget effect parameter.
`type`	type of spatial correlation function: `'exponential'`, `'gaussian'`, `'matern'`, and `'pow.exp'` for exponential, gaussian, matérn, and power exponential, respectively.
`kappa`	parameter for some spatial correlation functions. See `CovMat`.
`lower, upper`	vectors of lower and upper bounds for the optimization method. If unspecified, the default is `c(0.01,0.01)` for lower and `c(30,30)` for upper.
`MaxIter`	maximum number of iterations for the EM algorithm. By default `=300`.
`error`	maximum convergence error. By default `=1e-4`.
`show_se`	logical. It indicates if the standard errors should be estimated by default `=TRUE`.

Details

The spatial Gaussian model is given by

$Y = X\beta + \xi,$

where Y is the $n \times 1$ response vector, X is the $n \times q$ design matrix, β is the $q \times 1$ vector of regression coefficients to be estimated, and ξ is the error term, assumed to follow a normal distribution with zero-mean and covariance matrix $\Sigma = \sigma^2 R(\phi) + \tau^2 I_n$. We assume that Σ is non-singular and that X has a full rank (Diggle and Ribeiro 2007).

The estimation is carried out using the EM algorithm, originally proposed by Dempster et al. (1977). The conditional expectations required in the E-step are computed using the function `meanvarTMD` from the package `MomTrunc`.

Value

An object of class "sclm". Generic functions `print` and `summary` are available to display the fitted results. The `plot` method can be used to visualize convergence diagnostics of the parameter estimates.

Specifically, the following components are returned:

`Theta`	estimated parameters in all iterations, $\theta = (\beta, \sigma^2, \phi, \tau^2)$.
`theta`	final estimation of $\theta = (\beta, \sigma^2, \phi, \tau^2)$.
`beta`	estimated β.
`sigma2`	estimated σ^2.
`phi`	estimated ϕ.
`tau2`	estimated τ^2.
`EY`	first conditional moment computed in the last iteration.
`EYY`	second conditional moment computed in the last iteration.
`SE`	vector of standard errors of $\theta = (\beta, \sigma^2, \phi, \tau^2)$.
`InfMat`	observed information matrix.

loglik	log-likelihood for the EM method.
AIC	Akaike information criterion.
BIC	Bayesian information criterion.
Iter	number of iterations needed to converge.
time	processing time.
call	RcppCensSpatial call that produced the object.
tab	table of estimates.
critFin	selection criteria.
range	effective range.
ncens	number of censored/missing observations.
MaxIter	maximum number of iterations for the EM algorithm.

Note

The final EM estimates correspond to the parameter values obtained at the last iteration of the EM algorithm.

To fit a regression model for non-censored data, just set `ci` as a vector of zeros.

Author(s)

Katherine L. Valeriano, Christian E. Galarza, and Larissa A. Matos.

References

Dempster AP, Laird NM, Rubin DB (1977). “Maximum likelihood from incomplete data via the EM algorithm.” *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1), 1–38.

Diggle P, Ribeiro P (2007). *Model-based Geostatistics*. Springer.

See Also

MCEM.sclm, SAEM.sclm, predict.sclm

Examples

```
# Simulated example: 10% of left-censored observations
set.seed(1000)
n = 50   # Test with another values for n
coords = round(matrix(runif(2*n,0,15),n,2), 5)
x = cbind(rnorm(n), runif(n))
data = rCensSp(c(-1,3), 2, 4, 0.5, x, coords, "left", 0.10, 0, "gaussian")

fit = EM.sclm(y=data$y, x=x, ci=data$ci, lcl=data$lcl, ucl=data$ucl,
              coords=coords, phi0=3, nugget0=1, type="gaussian")
fit
```

MCEM.sclm *ML estimation of spatial censored linear models via the MCEM algorithm*

Description

It fits a spatial linear model with left-, right-, or interval-censored responses using the Monte Carlo EM (MCEM) algorithm. The function provides parameter estimates and their standard errors, and supports missing values in the response variable.

Usage

```
MCEM.sclm(y, x, ci, lcl = NULL, ucl = NULL, coords, phi0, nugget0,
  type = "exponential", kappa = NULL, lower = c(0.01, 0.01),
  upper = c(30, 30), MaxIter = 500, nMin = 20, nMax = 5000,
  error = 1e-04, show_se = TRUE)
```

Arguments

`y`	vector of responses of length n.
`x`	design matrix of dimensions $n \times q$, where q is the number of fixed effects, including the intercept.
`ci`	vector of censoring indicators of length n. For each observation: `1` if censored/missing, `0` otherwise.
`lcl, ucl`	vectors of length n representing the lower and upper bounds of the interval, which contains the true value of the censored observation. Default `=NULL`, indicating no-censored data. For each observation: `lcl=-Inf` and `ucl=c` (left censoring); `lcl=c` and `ucl=Inf` (right censoring); and `lcl` and `ucl` must be finite for interval censoring. Moreover, missing data could be defined by setting `lcl=-Inf` and `ucl=Inf`.
`coords`	2D spatial coordinates of dimensions $n \times 2$.
`phi0`	initial value for the spatial scaling parameter.
`nugget0`	initial value for the nugget effect parameter.
`type`	type of spatial correlation function: `'exponential'`, `'gaussian'`, `'matern'`, and `'pow.exp'` for exponential, gaussian, matérn, and power exponential, respectively.
`kappa`	parameter for some spatial correlation functions. See `CovMat`.
`lower, upper`	vectors of lower and upper bounds for the optimization method. If unspecified, the default is `c(0.01,0.01)` for lower and `c(30,30)` for upper.
`MaxIter`	maximum number of iterations for the MCEM algorithm. By default `=500`.
`nMin`	initial sample size for Monte Carlo integration. By default `=20`.
`nMax`	maximum sample size for Monte Carlo integration. By default `=5000`.
`error`	maximum convergence error. By default `=1e-4`.
`show_se`	logical. It indicates if the standard errors should be estimated by default `=TRUE`.

Details

The spatial Gaussian model is given by

$Y = X\beta + \xi,$

where Y is the $n \times 1$ response vector, X is the $n \times q$ design matrix, β is the $q \times 1$ vector of regression coefficients to be estimated, and ξ is the error term, assumed to follow a normal distribution with zero-mean and covariance matrix $\Sigma = \sigma^2 R(\phi) + \tau^2 I_n$. We assume that Σ is non-singular and that X has a full rank (Diggle and Ribeiro 2007).

Parameter estimation is carried out using the MCEM algorithm, originally proposed by Wei and Tanner (1990). The Monte Carlo (MC) approximation starts with a sample of size `nMin`. At each iteration, the sample size increases by $(nMax - nMin)/MaxIter$, reaching `nMax` at the final iteration. The random samples are generated using the slice sampling algorithm implemented in the `relliptical` package.

Value

An object of class "sclm". Generic functions `print` and `summary` are available to display the fitted results. The `plot` method can be used to visualize convergence diagnostics of the parameter estimates.

Specifically, the following components are returned:

`Theta`	estimated parameters in all iterations, $\theta = (\beta, \sigma^2, \phi, \tau^2)$.
`theta`	final estimation of $\theta = (\beta, \sigma^2, \phi, \tau^2)$.
`beta`	estimated β.
`sigma2`	estimated σ^2.
`phi`	estimated ϕ.
`tau2`	estimated τ^2.
`EY`	MC approximation of the first conditional moment.
`EYY`	MC approximation of the second conditional moment.
`SE`	vector of standard errors of $\theta = (\beta, \sigma^2, \phi, \tau^2)$.
`InfMat`	observed information matrix.
`loglik`	log-likelihood for the MCEM method.
`AIC`	Akaike information criterion.
`BIC`	Bayesian information criterion.
`Iter`	number of iterations needed to converge.
`time`	processing time.
`call`	`RcppCensSpatial` call that produced the object.
`tab`	table of estimates.
`critFin`	selection criteria.
`range`	effective range.
`ncens`	number of censored/missing observations.
`MaxIter`	maximum number of iterations for the MCEM algorithm.

Note

The MCEM final estimates correspond to the mean of the estimates obtained at each iteration after deleting the half and applying a thinning of 3.

To fit a regression model for non-censored data, just set `ci` as a vector of zeros.

Author(s)

Katherine L. Valeriano, Christian E. Galarza, and Larissa A. Matos.

References

Diggle P, Ribeiro P (2007). *Model-based Geostatistics*. Springer.

Wei G, Tanner M (1990). "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms." *Journal of the American Statistical Association*, **85**(411), 699–704. doi:10.1080/01621459.1990.10474930.

See Also

`EM.sclm`, `SAEM.sclm`, `predict.sclm`

Examples

```
# Example 1: left censoring data
set.seed(1000)
n = 50   # Test with another values for n
coords = round(matrix(runif(2*n,0,15),n,2), 5)
x = cbind(rnorm(n), rnorm(n))
data = rCensSp(c(2,-1), 2, 3, 0.70, x, coords, "left", 0.08, 0, "matern", 1)

fit = MCEM.sclm(y=data$y, x=x, ci=data$ci, lcl=data$lcl, ucl=data$ucl,
                coords, phi0=2.50, nugget0=0.75, type="matern",
                kappa=1, MaxIter=30, nMax=1000)
fit$tab

# Example 2: left censoring and missing data
yMiss = data$y
yMiss[20] = NA
ci = data$ci
ci[20] = 1
ucl = data$ucl
ucl[20] = Inf

fit1 = MCEM.sclm(y=yMiss, x=x, ci=ci, lcl=data$lcl, ucl=ucl, coords,
                 phi0=2.50, nugget0=0.75, type="matern", kappa=1,
                 MaxIter=300, nMax=1000)
summary(fit1)
plot(fit1)
```

`Missouri` *TCDD concentration data*

Description

The level of dioxin (2,3,7,8-tetrachlorodibenzo-p-dioxin or TCDD) data was collected in November 1983 by the U.S. Environmental Protection Agency (EPA) in several areas of a highway in Missouri, USA. The TCDD measurement was subject to a limit of detection (`cens`); thereby, the `TCDD` data is left-censored. Only the locations used in the geostatistical analysis by Zirschky and Harris (1986) are shown.

Usage

```
data("Missouri")
```

Format

A data frame with 127 observations and five variables:

xcoord x coordinate of the start of each transect (ft).

ycoord y coordinate of the start of each transect (ft).

TCDD TCDD concentrations (mg/kg).

transect transect length (ft).

cens indicator of censoring (left-censored observations).

Source

Zirschky JH, Harris DJ (1986). "Geostatistical analysis of hazardous waste site data." *Journal of Environmental Engineering*, **112**(4), 770–784.

See Also

`EM.sclm`, `MCEM.sclm`, `SAEM.sclm`

Examples

```
data("Missouri")
y = log(Missouri$TCDD)
cc = Missouri$cens
coord = cbind(Missouri$xcoord/100, Missouri$ycoord)
x = matrix(1, length(y), 1)
lcl = rep(-Inf, length(y))
ucl = y

## SAEM fit
set.seed(83789)
fit1 = SAEM.sclm(y, x, cc, lcl, ucl, coord, 5, 1, lower=c(1e-5,1e-5),
                 upper=c(50,50))
```

```
fit1$tab

## MCEM fit
fit2 = MCEM.sclm(y, x, cc, lcl, ucl, coord, 5, 1, lower=c(1e-5,1e-5),
                 upper=c(50,50), MaxIter=300, nMax=1000)
fit2$tab

## Imputed values
cbind(fit1$EY, fit2$EY)[cc==1,]
```

`predict.sclm` *Prediction in spatial models with censored/missing responses*

Description

It performs spatial prediction at a set of new `S` spatial locations.

Usage

```
## S3 method for class 'sclm'
predict(object, locPre, xPre, ...)
```

Arguments

`object`	object of class 'sclm' given as output of `EM.sclm`, `MCEM.sclm`, or `SAEM.sclm` function.
`locPre`	matrix of coordinates for which prediction is performed.
`xPre`	matrix of covariates for which prediction is performed.
`...`	further arguments passed to or from other methods.

Details

This function performs prediction under the mean squared error (MSE) criterion, where the conditional expectation $E(Y|X)$ is used as the optimal predictor.

Value

The function returns a list with:

`coord`	matrix of coordinates.
`predValues`	predicted values.
`sdPred`	predicted standard deviations.

Author(s)

Katherine L. Valeriano, Christian E. Galarza, and Larissa A. Matos.

See Also

EM.sclm, MCEM.sclm, SAEM.sclm

Examples

```
set.seed(1000)
n = 120
coords = round(matrix(runif(2*n,0,15),n,2), 5)
x = cbind(rbinom(n,1,0.50), rnorm(n), rnorm(n))
data = rCensSp(c(1,4,-1), 2, 3, 0.50, x, coords, "left", 0.10, 20)

## Estimation
data1 = data$Data

# Estimation: EM algorithm
fit1 = EM.sclm(y=data1$y, x=data1$x, ci=data1$ci, lcl=data1$lcl,
               ucl=data1$ucl, coords=data1$coords, phi0=2.50, nugget0=1)

# Estimation: SAEM algorithm
fit2 = SAEM.sclm(y=data1$y, x=data1$x, ci=data1$ci, lcl=data1$lcl,
                 ucl=data1$ucl, coords=data1$coords, phi0=2.50, nugget0=1)

# Estimation: MCEM algorithm
fit3 = MCEM.sclm(y=data1$y, x=data1$x, ci=data1$ci, lcl=data1$lcl,
                 ucl=data1$ucl, coords=data1$coords, phi0=2.50, nugget0=1,
                 MaxIter=300)
cbind(fit1$theta, fit2$theta, fit3$theta)

# Prediction
data2 = data$TestData
pred1 = predict(fit1, data2$coords, data2$x)
pred2 = predict(fit2, data2$coords, data2$x)
pred3 = predict(fit3, data2$coords, data2$x)

# Cross-validation
mean((data2$y - pred1$predValues)^2)
mean((data2$y - pred2$predValues)^2)
mean((data2$y - pred3$predValues)^2)
```

---

rCensSp *Censored spatial data simulation*

Description

It simulates censored spatial data under a linear model for a specified censoring rate.

Usage

```
rCensSp(beta, sigma2, phi, nugget, x, coords, cens = "left", pcens = 0.1,
  npred = 0, cov.model = "exponential", kappa = NULL)
```

Arguments

beta	linear regression parameters.
sigma2	partial sill parameter.
phi	spatial scaling parameter.
nugget	nugget effect parameter.
x	design matrix of dimensions $n \times q$.
coords	2D spatial coordinates of dimensions $n \times 2$.
cens	'left' or 'right' censoring. By default ='left'.
pcens	desired censoring rate. By default =0.10.
npred	number of simulated data used for cross-validation (Prediction). By default =0.
cov.model	type of spatial correlation function: 'exponential', 'gaussian', 'matern', and 'pow.exp' for exponential, gaussian, matérn, and power exponential, respectively.
kappa	parameter for some spatial correlation functions. For exponential and gaussian kappa=NULL, for power exponential 0 < kappa <= 2, and for matérn correlation function kappa > 0.

Value

If npred > 0, it returns two lists: Data and TestData; otherwise, it returns a list with the simulated data.

Data

y	response vector.
ci	censoring indicator.
lcl	lower censoring bound.
ucl	upper censoring bound.
coords	coordinates matrix.
x	design matrix.

TestData

y	response vector.
coords	coordinates matrix.
x	design matrix.

Author(s)

Katherine L. Valeriano, Christian E. Galarza, and Larissa A. Matos.

Examples

```
set.seed(1000)
n = 100
coords = round(matrix(runif(2*n,0,15),n,2), 5)
x = cbind(1, rnorm(n))
data = rCensSp(beta=c(5,2), sigma2=2, phi=4, nugget=0.70, x=x,
               coords=coords, cens="left", pcens=0.10, npred=10,
               cov.model="gaussian")
data$Data
data$TestData
```

SAEM.sclm — *ML estimation of spatial censored linear models via the SAEM algorithm*

Description

It fits a spatial linear model with left-, right-, or interval-censored responses using the Stochastic Approximation EM (SAEM) algorithm. The function provides parameter estimates and their standard errors, and supports missing values in the response variable.

Usage

```
SAEM.sclm(y, x, ci, lcl = NULL, ucl = NULL, coords, phi0, nugget0,
  type = "exponential", kappa = NULL, lower = c(0.01, 0.01),
  upper = c(30, 30), MaxIter = 300, M = 20, pc = 0.2, error = 1e-04,
  show_se = TRUE)
```

Arguments

`y`	vector of responses of length n.
`x`	design matrix of dimensions $n \times q$, where q is the number of fixed effects, including the intercept.
`ci`	vector of censoring indicators of length n. For each observation: `1` if censored/missing, `0` otherwise.
`lcl`, `ucl`	vectors of length n representing the lower and upper bounds of the interval, which contains the true value of the censored observation. Default `=NULL`, indicating no-censored data. For each observation: `lcl=-Inf` and `ucl=c` (left censoring); `lcl=c` and `ucl=Inf` (right censoring); and `lcl` and `ucl` must be finite for interval censoring. Moreover, missing data could be defined by setting `lcl=-Inf` and `ucl=Inf`.
`coords`	2D spatial coordinates of dimensions $n \times 2$.
`phi0`	initial value for the spatial scaling parameter.
`nugget0`	initial value for the nugget effect parameter.

type	type of spatial correlation function: 'exponential', 'gaussian', 'matern', and 'pow.exp' for exponential, gaussian, matérn, and power exponential, respectively.
kappa	parameter for some spatial correlation functions. See CovMat.
lower, upper	vectors of lower and upper bounds for the optimization method. If unspecified, the default is c(0.01,0.01) for lower and c(30,30) for upper.
MaxIter	maximum number of iterations of the SAEM algorithm. By default =300.
M	number of Monte Carlo samples for stochastic approximation. By default =20.
pc	percentage of initial iterations of the SAEM algorithm with no memory. It is recommended that 50<MaxIter*pc<100. By default =0.20.
error	maximum convergence error. By default =1e-4.
show_se	logical. It indicates if the standard errors should be estimated by default =TRUE.

Details

The spatial Gaussian model is given by

$Y = X\beta + \xi,$

where Y is the $n \times 1$ response vector, X is the $n \times q$ design matrix, β is the $q \times 1$ vector of regression coefficients to be estimated, and ξ is the error term, assumed to follow a normal distribution with zero-mean and covariance matrix $\Sigma = \sigma^2 R(\phi) + \tau^2 I_n$. We assume that Σ is non-singular and that X has a full rank (Diggle and Ribeiro 2007).

Parameter estimation is carried out using the SAEM algorithm, originally proposed by Delyon et al. (1999). The spatial censored SAEM approach has been previously developed by Lachos et al. (2017) and Ordoñez et al. (2018), and is implemented in the CensSpatial package. Differences among implementations mainly arise from the random number generation schemes and optimization procedures.

This model can also be viewed as a particular case of the spatio-temporal model proposed by Valeriano et al. (2021), when the number of temporal observations is equal to one. The corresponding SAEM implementation for the spatio-temporal setting is available in the StempCens package.

Value

An object of class "sclm". Generic functions print and summary are available to display the fitted results. The plot method can be used to visualize convergence diagnostics of the parameter estimates.

Specifically, the following components are returned:

Theta	estimated parameters in all iterations, $\theta = (\beta, \sigma^2, \phi, \tau^2)$.
theta	final estimation of $\theta = (\beta, \sigma^2, \phi, \tau^2)$.
beta	estimated β.
sigma2	estimated σ^2.
phi	estimated ϕ.
tau2	estimated τ^2.
EY	stochastic approximation of the first conditional moment.

EYY	stochastic approximation of the second conditional moment.
SE	vector of standard errors of $\theta = (\beta, \sigma^2, \phi, \tau^2)$.
InfMat	observed information matrix.
loglik	log-likelihood for the SAEM method.
AIC	Akaike information criterion.
BIC	Bayesian information criterion.
Iter	number of iterations needed to converge.
time	processing time.
call	RcppCensSpatial call that produced the object.
tab	table of estimates.
critFin	selection criteria.
range	effective range.
ncens	number of censored/missing observations.
MaxIter	maximum number of iterations for the SAEM algorithm.

Note

The SAEM final estimates correspond to the estimates obtained at the last iteration of the algorithm.

To fit a regression model for non-censored data, just set ci as a vector of zeros.

Author(s)

Katherine L. Valeriano, Christian E. Galarza, and Larissa A. Matos.

References

Delyon B, Lavielle M, Moulines E (1999). "Convergence of a stochastic approximation version of the EM algorithm." *The Annals of Statistics*, **27**(1), 94–128.

Diggle P, Ribeiro P (2007). *Model-based Geostatistics*. Springer.

Lachos VH, Matos LA, Barbosa TS, Garay AM, Dey DK (2017). "Influence diagnostics in spatial models with censored response." *Environmetrics*, **28**(7).

Ordoñez JA, Bandyopadhyay D, Lachos VH, Cabral CRB (2018). "Geostatistical estimation and prediction for censored responses." *Spatial Statistics*, **23**, 109–123. doi:10.1016/j.spasta.2017.12.001.

Valeriano KL, Lachos VH, Prates MO, Matos LA (2021). "Likelihood-based inference for spatiotemporal data with censored and missing responses." *Environmetrics*, **32**(3).

See Also

EM.sclm, MCEM.sclm, predict.sclm

Examples

```
# Example 1: 8% of right-censored observations
set.seed(1000)
n = 50   # Test with another values for n
coords = round(matrix(runif(2*n,0,15),n,2), 5)
x = cbind(rnorm(n), rnorm(n))
data = rCensSp(c(4,-2), 1, 3, 0.50, x, coords, "right", 0.08)

fit = SAEM.sclm(y=data$y, x=x, ci=data$ci, lcl=data$lcl, ucl=data$ucl,
                coords, phi0=2, nugget0=1, type="exponential", M=10,
                pc=0.18)
fit

# Example 2: censored and missing observations
set.seed(123)
n = 200
coords = round(matrix(runif(2*n,0,20),n,2), 5)
x = cbind(runif(n), rnorm(n), rexp(n))
data = rCensSp(c(1,4,-1), 2, 3, 0.50, x, coords, "left", 0.05, 0,
               "matern", 3)
data$y[c(10,120)] = NA
data$ci[c(10,120)] = 1
data$ucl[c(10,120)] = Inf

fit2 = SAEM.sclm(y=data$y, x=x, ci=data$ci, lcl=data$lcl, ucl=data$ucl,
                 coords, phi0=2, nugget0=1, type="matern", kappa=3,
                 M=10, pc=0.18)
fit2$tab
plot(fit2)
```

Index