Package ‘RcppCensSpatial’

September 21, 2021

Type Package

Title Spatial Estimation and Prediction for Censored/Missing Responses

Description It provides functions to estimate the parameters in spatial models with censored/missing responses via the Expectation-Maximization (EM) algorithm (see Dempster, Laird, and Rubin (1977)<https://www.jstor.org/stable/2984875>), the Stochastic Approximation EM (SAEM) algorithm (see Delyon, Lavielle, and Moulines (1999)<https://www.jstor.org/stable/120128>), and the Monte Carlo EM (MCEM) algorithm (see Wei and Tanner (1990)<doi:10.1080/01621459.1990.10474930>). These algorithms are widely used to compute the maximum likelihood (ML) estimates in incomplete data problems. The EM algorithm computes the ML estimates when a closed expression for the conditional expectation of the complete-data log-likelihood function is available. In the MCEM algorithm, the conditional expectation is substituted by a Monte Carlo approximation based on many independent simulations of the missing data, while the SAEM algorithm splits the E-step into a simulation step and an integration step. The SAEM algorithm was developed as an alternative to the computationally intensive MCEM algorithm. This package also approximates the standard error of the estimates using the method developed by Louis (1982)<https://www.jstor.org/stable/2345828>. It also has a function that performs spatial prediction in a set of new locations. Besides the functions to estimate parameters, this package allows computing the covariance matrix and the distance matrix.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.1.1

Imports ggplot2, gridExtra, MomTrunc, mvtnorm, Rcpp, Rdpack, stats, tlmvnmvt

RdMacros Rdpack

LinkingTo RcppArmadillo, Rcpp, RcppProgress, roptim

Suggests CensSpatial,relliptical, StempCens

NeedsCompilation yes

Author Katherine A. L. Valeriano [aut, cre]
(https://orcid.org/0000-0001-6388-4753), Alejandro Ordonez Cuastumal [ctb]
Covariance Matrix for Spatial Models

Description

This function computes the spatial variance-covariance matrix considering exponential, gaussian, matern, or power exponential correlation functions.

Usage

CovMat(phi, tau2, sigma2, dist, type = "exponential", kappa = 0)

Arguments

phi  
spatial scaling parameter.

tau2  
nugget effect parameter.

sigma2  
partial sill parameter.

dist  
nxn distance matrix.

type  
type of spatial correlation function: 'exponential', 'gaussian', 'matern', and 'pow.exp' for exponential, gaussian, matern, and power exponential, respectively.

kappa  
parameter for all spatial correlation functions. For exponential and gaussian \( \kappa = 0 \), for power exponential \( 0 < \kappa \leq 2 \), and for matern correlation function \( \kappa > 0 \).
**CovMat**

**Details**

The spatial covariance matrix is given by

\[
\Sigma = \left[ \text{Cov}(s_i, s_j) \right] = \sigma^2 R(\phi) + \tau^2 I_n,
\]

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

The spatial correlation functions available are:

**Exponential:** \(\text{Corr}(d) = \exp(-d/\phi)\),

**Gaussian:** \(\text{Corr}(d) = \exp(-(d/\phi)^2)\),

**Matern:** \(\text{Corr}(d) = 1/(2^\kappa - 1)\Gamma(\kappa)(d/\phi)^\kappa K_\kappa(d/\phi)\),

**Power exponential:** \(\text{Corr}(d) = \exp(-(d/\phi)^\kappa)\),

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

**Value**

The function returns the \(nxn\) spatial covariance matrix.

**Author(s)**

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

**See Also**

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

**Examples**

```r
# Initial parameter values
phi = 5; tau2 = 0.80; sigma2 = 2
n = 20
set.seed(1000)
x = round(runif(n,0,10), 5)  # X coordinate
y = round(runif(n,0,10), 5)  # Y coordinate
Ms = dist2Dmatrix(cbind(x, y))
Cov = CovMat(phi, tau2, sigma2, Ms, "exponential", 0)
```
### dist2Dmatrix  
*Distance Matrix Computation*

**Description**
This function computes the Euclidean distance matrix for a set of coordinates.

**Usage**
```r
dist2Dmatrix(coords)
```

**Arguments**
- `coords`: 2D spatial coordinates.

**Value**
The function returns the $n \times n$ distance matrix.

**Author(s)**
Katherine L. Valeriano, Alejandro Ordonez, Christian E. Galarza and Larissa A. Matos.

**Examples**
```r
n = 100
generate.seed(1000)
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  
*Censored Spatial Model Estimation via EM Algorithm*

**Description**
This function returns the maximum likelihood (ML) estimates of the unknown parameters in Gaussian spatial models with censored/missing responses via the EM algorithm. It supports left, right, interval, or missing values in the dependent variable. It also computes the observed information matrix using the method developed by Louis (1982).

**Usage**
```r
EM.sclm(y, x, cens, LI, LS, coords, init.phi, init.nugget,
type = "exponential", kappa = 0, lower = c(0.01, 0.01), upper = c(30, 30), MaxIter = 300, error = 1e-05, show.SE = TRUE)
```
Arguments

- `y`: vector of responses.
- `x`: design matrix.
- `cens`: vector of censoring indicators. For each observation: 1 if censored/missing and 0 otherwise.
- `LI`: lower limit of detection. For each observation: if non-censored =y, if left-censored/missing =-Inf, or =LOD if right/interval-censored.
- `LS`: upper limit of detection. For each observation: if non-censored =y, if right-censored/missing =Inf, or =LOD if left/interval-censored.
- `coords`: 2D spatial coordinates.
- `init.phi`: initial value for the spatial scaling parameter.
- `init.nugget`: initial value for the nugget effect parameter.
- `type`: type of spatial correlation function: 'exponential', 'gaussian', 'matern', and 'pow.exp' for exponential, gaussian, matern, and power exponential, respectively.
- `kappa`: parameter for all 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 EM algorithm. By default =300.
- `error`: maximum convergence error. By default =1e-5.
- `show.SE`: TRUE or FALSE. 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 \( nx1 \) vector of response, \( X \) is the \( nxq \) design matrix, \( \beta \) is the \( qx1 \) vector of regression coefficients to be estimated, and \( \xi \) is the error term which is normally distributed with zero-mean and covariance matrix \( \Sigma = \sigma^2 R(\phi) + \tau^2 I_n \). We assume that \( \Sigma \) is non-singular and \( X \) has full rank (Diggle and Ribeiro 2007).

The estimation process was performed via the EM algorithm initially proposed by Dempster et al. (1977). The conditional expectations are computed through function meanvarTMD available in package MomTrunc (Galarza et al. 2019).

Value

The function returns an object of class `sclm` which is a list given by:

- `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 \( \beta \).
- `sigma2`: estimated \( \sigma^2 \).
phi estimated $\phi$.

tau2 estimated $\tau^2$.

EY first moment for the truncated normal distribution computed in the last iteration.

EYY second moment for the truncated normal distribution 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.

Iterations number of iterations needed to converge.

ptime processing time.

range the effective range.

Note

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

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

Functions print, summary, and plot work for objects of class sclm.

Author(s)

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

References


See Also

SAEM.sclm, MCEM.sclm, predict.sclm
Examples

# Simulated example: 10% of left-censored observations
n = 50      # Test with another values for n
set.seed(1000)
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",0)

fit = EM.sclm(y=data$yobs, x=data[,7:8], cens=data$cens, LI=data$LI,
LS=data$LS, coords=data[,5:6], init.phi=3, init.nugget=1,
type="gaussian", error=1e-4)

fit

MCEM.sclm Censored Spatial Model Estimation via MCEM Algorithm

Description

This function returns the maximum likelihood (ML) estimates of the unknown parameters in Gaussian spatial models with censored/missing responses via the MCEM algorithm. It supports left, right, interval, or missing values in the dependent variable. It also computes the observed information matrix using the method developed by Louis (1982).

Usage

MCEM.sclm(y, x, cens, LI, LS, coords, init.phi, init.nugget,
type = "exponential", kappa = 0, lower = c(0.01, 0.01), upper = c(30, 30), MaxIter = 500, nMin = 20, nMax = 5000, error = 1e-05,
show.SE = TRUE)

Arguments

y      vector of responses.
x      design matrix.
cens   vector of censoring indicators. For each observation: 1 if censored/missing and 0 otherwise.
LI     lower limit of detection. For each observation: if non-censored =y, if left-censored/missing =-Inf, or =LOD if right/interval-censored.
LS     upper limit of detection. For each observation: if non-censored =y, if right-censored/missing =Inf, or =LOD if left/interval-censored.
coords 2D spatial coordinates.
init.phi initial value for the spatial scaling parameter.
init.nugget initial value for the nugget effect parameter.
type type of spatial correlation function: 'exponential', 'gaussian', 'matern', and 'pow.exp' for exponential, gaussian, matern, and power exponential, respectively.
kappa parameter for all 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 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-5.
show.SE TRUE or FALSE. 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 \( nx1 \) vector of response, \( X \) is the \( nxq \) design matrix, \( \beta \) is the \( qx1 \) vector of regression coefficients to be estimated, and \( \xi \) is the error term which is normally distributed with zero-mean and covariance matrix \( \Sigma = \sigma^2 R(\phi) + \tau^2 I_n \). We assume that \( \Sigma \) is non-singular and \( X \) has full rank (Diggle and Ribeiro 2007).

The estimation process was performed via the MCEM algorithm initially proposed by Wei and Tanner (1990). The Monte Carlo integration starts with a sample of size \( n\text{Min} \); at each iteration, the sample size increases \((n\text{Max}-n\text{Min})/\text{MaxIter}\), and at the last iteration, the sample size is \( n\text{Max} \). The random observations are sampled through the slice sampling algorithm available in package relliptical.

Value

The function returns an object of class sclm which is a list given by:

- 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 \( \beta \).
- sigma2 estimated \( \sigma^2 \).
- phi estimated \( \phi \).
- tau2 estimated \( \tau^2 \).
- EY MC approximation of the first moment for the truncated normal distribution.
- EYY MC approximation of the second moment for the truncated normal distribution.
- 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.
- Iterations number of iterations needed to converge.
- ptime processing time.
- range the effective range.
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 cens as a vector of zeros.

Functions print, summary, and plot work for objects of class sclm.

Author(s)

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

References


See Also

EM.sclm, SAEM.sclm, predict.sclm

Examples

# Simulated example: censored and missing data
n = 50  # Test with another values for n
set.seed(1000)
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)
data$yobs[20] = NA
fit = MCEM.sclm(y=data$yobs, x=data[,7:8], cens=data$cens, LI=data$LI, LS=data$LS, coords=data[,5:6], init.phi=2.50, init.nugget=0.75, type="matern", kappa=1, MaxIter=20, nMax=1000, error=1e-4)
print(fit)

# Application: TCDD concentration in Missouri
library(CensSpatial)
data("Missouri")
y = log(Missouri$V3)
cc = Missouri$V5
coord = cbind(Missouri$V1/100,Missouri$V2)
X = matrix(1,length(y),1)
predict.sclm

LI = LS = y; LI[cc==1] = -Inf

fit2 = MCEM.sclm(y=y, x=X, cens=cc, LI=LI, LS=LS, coords=coord, init.phi=5,
init.nugget=1, type="exponential", lower=c(1e-5,1e-5), upper=c(50,50),
MaxIter=500, nMax=1000, error=1e-5)

summary(fit2)
plot(fit2)

predict.sclm  Prediction in Spatial Model with censored/missing responses

Description
This function performs spatial prediction in 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, SAEM.sclm or MCEM.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 predicts using the Mean Squared Error (MSE) criterion, which takes the conditional
expectation E(Y|X) as the best linear predictor.

Value
The function returns a data frame with:
  xcoord        x coordinates.
  ycoord        y coordinates.
  predValues    predicted values.
  sdPred        predicted standard deviations.

Author(s)
Katherine L. Valeriano, Alejandro Ordonez, Christian E. Galarza and Larissa A. Matos.
Censored Spatial Data Simulation

Description

This function simulates censored spatial data with a linear structure for an established censoring rate.
Usage

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

Arguments

- **beta**: linear regression parameters.
- **sigma2**: partial sill parameter.
- **phi**: spatial scaling parameter.
- **nugget**: nugget effect parameter.
- **x**: design matrix.
- **coords**: 2D spatial coordinates.
- **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, matern, and power exponential, respectively.
- **kappa**: parameter for all spatial correlation functions. For exponential and gaussian $\kappa = 0$, for power exponential $0 < \kappa \leq 2$, and for matern correlation function $\kappa > 0$.

Value

If $\text{npred} > 0$, it returns a list with two datasets: TrainingData and TestData; otherwise, it returns a data frame with the simulated data.

**TrainingData**

- **yobs**: generated response vector.
- **cens**: censoring indicator.
- **LI**: lower censoring bound.
- **LS**: upper censoring bound.
- **xcoord**: x coordinates.
- **ycoord**: y coordinates.
- **X**: design matrix.

**TestData**

- **yobs**: generated response vector.
- **xcoord**: x coordinates.
- **ycoord**: y coordinates.
- **X**: design matrix.
**Author(s)**

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

**Examples**

```r
n = 100
set.seed(1000)
coords = round(matrix(runif(2*n,0,15),n,2),5)
x = cbind(1, rnorm(n))
data = rCensSp(c(5,2),2,4,0.70,x,coords,"left",0.10,10,"gaussian",0)
data$TrainingData
data$TestData
```

**SAEM.sclm**

Censored Spatial Model Estimation via SAEM Algorithm

**Description**

This function returns the maximum likelihood (ML) estimates of the unknown parameters in Gaussian spatial models with censored/missing responses via the SAEM algorithm. It supports left, right, interval, or missing values in the dependent variable. It also computes the observed information matrix using the method developed by Louis (1982).

**Usage**

```r
SAEM.sclm(y, x, cens, LI, LS, coords, init.phi, init.nugget,
  type = "exponential", kappa = 0, lower = c(0.01, 0.01), upper = c(30, 30), MaxIter = 300, M = 20, pc = 0.25, error = 1e-05,
  show.SE = TRUE)
```

**Arguments**

- `y` vector of responses.
- `x` design matrix.
- `cens` vector of censoring indicators. For each observation: 1 if censored/missing and 0 otherwise.
- `LI` lower limit of detection. For each observation: if non-censored =y, if left-censored/missing =-Inf, or =LOD if right/interval-censored.
- `LS` upper limit of detection. For each observation: if non-censored =y, if right-censored/missing =Inf, or =LOD if left/interval-censored.
- `coords` 2D spatial coordinates.
- `init.phi` initial value for the spatial scaling parameter.
- `init.nugget` initial value for the nugget effect parameter.
- `type` type of spatial correlation function: 'exponential', 'gaussian', 'matern', and 'pow.exp' for exponential, gaussian, matern, and power exponential, respectively.
The spatial Gaussian model is given by
\[ Y = X\beta + \xi, \]
where \( Y \) is the \( nx1 \) vector of response, \( X \) is the \( nxq \) design matrix, \( \beta \) is the \( qx1 \) vector of regression coefficients to be estimated, and \( \xi \) is the error term which is normally distributed with zero-mean and covariance matrix \( \Sigma = \sigma^2 R(\phi) + \tau^2 I_n \). We assume that \( \Sigma \) is non-singular and \( X \) has full rank (Diggle and Ribeiro 2007).

The estimation process was performed via the SAEM (Delyon et al. 1999) algorithm. The spatial SAEM algorithm was previously proposed by Lachos et al. (2017) and Ordonez et al. (2018) and is available in package CensSpatial. The difference between this package to CensSpatial is that the random observations are sampled through the slice sampling algorithm available in package relliptical and the optimization procedure by the roptim package.

This model is also a particular case of the Spatio-temporal model defined by Valeriano et al. (2020), when the number of temporal observations is equal to one. The computing codes of the Spatio-temporal SAEM algorithm are available in the package StempCens.

Value
The function returns an object of class sclm which is a list given by:

\[
\begin{align*}
\text{Theta} & \quad \text{estimated parameters in all iterations, } \theta = (\beta, \sigma^2, \phi, \tau^2). \\
\text{theta} & \quad \text{final estimation of } \theta = (\beta, \sigma^2, \phi, \tau^2). \\
\text{beta} & \quad \text{estimated } \beta. \\
\text{sigma2} & \quad \text{estimated } \sigma^2. \\
\text{phi} & \quad \text{estimated } \phi. \\
\text{tau2} & \quad \text{estimated } \tau^2. \\
\text{EY} & \quad \text{stochastic approximation of the first moment for the truncated normal distribution.} \\
\text{EYY} & \quad \text{stochastic approximation of the second moment for the truncated normal distribution.} \\
\text{SE} & \quad \text{vector of standard errors of } \theta = (\beta, \sigma^2, \phi, \tau^2). 
\end{align*}
\]
InfMat observed information matrix.
loglik log-likelihood for the SAEM method.
AIC Akaike information criterion.
BIC Bayesian information criterion.
Iterations number of iterations needed to converge.
pTime processing time.
range the effective range.

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 cens as a vector of zeros.
Functions print, summary, and plot work for objects of class sclm.

Author(s)
Katherine L. Valeriano, Alejandro Ordonez, Christian E. Galarza and Larissa A. Matos.

References


See Also
EM.sclm MCEM.sclm predict.sclm
Examples

# Simulated example: 10% of right-censored observations
n = 50  # Test with another values for n
set.seed(1000)
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,-2),2,3,0.50,x,coords,"right",0.10,0,"matern",2)

fit = SAEM.sclm(y=data$yobs, x=data[,7:9], cens=data$cens, LI=data$LI,
LS=data$LS, coords=data[,5:6], init.phi=2, init.nugget=1,
type="matern", kappa=2, MaxIter = 20, error=1e-4)

summary(fit)

# Simulated example: censored and missing observations
n = 200
set.seed(123)
coords = round(matrix(runif(2*n,0,20),n,2),5)
x = cbind(1, rnorm(n), rexp(n))
data = rCensSp(c(1,4,-1),2,4,0.50,x,coords,"left",0.10,0,"exponential",0)
data$yobs[c(10,20)] = NA; data$cens[c(10,20)] = 1
data$LI[c(10,20)] = -Inf; data$LS[c(10,20)] = Inf

fit2 = SAEM.sclm(y=data$yobs, x=data[,7:9], cens=data$cens, LI=data$LI,
LS=data$LS, coords=data[,5:6], init.phi=2, init.nugget=1,
type="exponential", MaxIter = 300, error=1e-4)

fit2$theta  # Estimates
fit2$SE  # Standard error
fit2$InfMat  # Information matrix
plot(fit2)
Index

CovMat, 2, 5, 8, 14

dist2Dmatrix, 3, 4

EM.sclm, 3, 4, 9–11, 15

MCEM.sclm, 3, 6, 7, 10, 11, 15

predict.sclm, 6, 9, 10, 15

rCensSp, 11

SAEM.sclm, 3, 6, 9–11, 13