Package: CensSpatial (via r-universe)

October 11, 2024

Type Package

Title Censored Spatial Models

Version 3.6

Depends R (>= 4.1.0)

Date 2023-01-18

Author Alejandro Ordonez, Christian E. Galarza, Victor H. Lachos

Maintainer Alejandro Ordonez <ordonezjosealejandro@gmail.com>

Imports geoR (>= 1.8-1), Rcpp, stats, graphics, mvtnorm, optimx (>= 2021.10-12), tmvtnorm (>= 1.4-10), msm, psych, numDeriv (>= 2.11.1),raster,moments (>= 0.14),lattice, tlrmvnmvt (>= 1.1.0)

Description It fits linear regression models for censored spatial data. It provides different estimation methods as the SAEM (Stochastic Approximation of Expectation Maximization) algorithm and seminaive that uses Kriging prediction to estimate the response at censored locations and predict new values at unknown locations. It also offers graphical tools for assessing the fitted model. More details can be found in Ordonez et al. (2018) <doi:10.1016/j.spasta.2017.12.001>.

Repository https://joalor93.r-universe.dev

RemoteUrl https://github.com/joalor93/censspatial

RemoteRef HEAD

RemoteSha eda79c90e9dc4cdff744ad1e55aa34046c10529c

Contents

`algnaive12` *Naive 1 and Naive 2 method for spatial prediction.*

Description

This function performs spatial censored estimation and prediction for left and right censure through the Naive 1 and Naive 2 methods.

Usage

```
algnaive12(data, cc, copred, thetaini, y.col = 3,coords.col = 1:2,covar=FALSE, covar.col,
fix.nugget = TRUE, nugget, kappa = 0, cutoff, cov.model = "exponential", trend)
```

Arguments

`data`	data.frame containing the coordinates, covariates and the response variable (in any order).
`cc`	(binary vector) indicator of censure (1: censored observation 0: observed).
`copred`	coordinates used in the prediction procedure.
`thetaini`	initial values for the σ^2 and ϕ values in the covariance structure.
`y.col`	(numeric) column of data.frame that corresponds to the response variable.
`coords.col`	(numeric) columns of data.frame that corresponds to the coordinates of the spatial data.
`covar`	(logical) indicates the presence of covariates in the spatial censored estimation (`FALSE` :without covariates, `TRUE` :with covariates).
`covar.col`	(numeric) columns of data.frame that corresponds to the covariates in the spatial censored linear model estimation.
`fix.nugget`	(logical) it indicates if the τ^2 parameter must be fixed.
`nugget`	(numeric) values of the τ^2 parameter, if `fix.nugget=F` this value corresponds to an initial value.
`kappa`	value of κ used in some covariance functions.
`cutoff`	(vector) Limit of censure detection (rc:>cutoff, lc:<cutoff).

cov.model	structure of covariance (see cov.spatial from geoR).
trend	it specifies the mean part of the model. See documentation of trend.spatial from geoR for further details. By default it takes "cte".

Details

The Naive 1 and Naive 2 are computed as in Schelin (2014). The naive 1 replaces the censored observations by the limit of detection (LD) and it performs estimation and prediction with this data. Instead of 1, the naive 2 replaces the censored observations by LD/2.

Value

beta1	beta parameter for the mean structure in the Naive 1 method.
beta2	beta parameter for the mean structure in the Naive 2 method.
theta1	vector of estimate parameter for the mean and covariance structure $(\beta, \sigma^2, \phi, \tau^2)$ in the Naive 1 method.
theta2	vector of estimate parameter for the mean and covariance structure $(\beta, \sigma^2, \phi, \tau^2)$ in the Naive 2 method.
predictions1	predictions obtained for the Naive 1 method.
predictions2	predictions obtained for the Naive 2 method.
AIC1	AIC of the estimated model in the Naive 1 method.
AIC2	AIC of the estimated model in the Naive 2 method.
BIC1	BIC of the estimated model in the Naive 1 method.
BIC2	BIC of the estimated model in the Naive 2 method.
loglik1	log likelihood for the estimated model in the Naive 1 method.
loglik2	log likelihood for the estimated model in the Naive 2 method.
sdpred1	standard deviations of predictions in the Naive 1 method.
sdpred2	standard deviations of predictions in the Naive 2 method.
type	covariance function used in estimation.
trend1	trend form for the mean structure.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

Schelin, L. & Sjostedt-de Luna, S. (2014). Spatial prediction in the presence of left-censoring. Computational Statistics and Data Analysis, 74.

See Also

SAEMSCL

Examples

```
###simulated coordinates
n<-200 ### sample size for estimation.
n1=100 ### number of observation used in the prediction.

###simulated coordinates
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)### total coordinates (used in estimation and prediction).

coords1=coords[1:n,]####coordinates used for estimation.

type="matern"### covariance structure.

xtot<-cbind(1,runif((n+n1)),runif((n+n1),2,3))## X matrix for estimation and prediction.
xobs=xtot[1:n,]## X matrix for estimation.

###simulated data
obj=rspacens(cov.pars=c(3,.3,0),beta=c(5,3,1),x=xtot,coords=coords,kappa=1.2,
cens=0.25,n=(n+n1),n1=n1,cov.model=type,cens.type="left")

data2=obj$datare
data2[,4:5]=xobs[,-1]

cc=obj$cc
y=obj$datare[,3]
cutoff=rep(obj$cutoff,length(y[cc==1]))


aux2=algnaive12(data=data2,cc=obj$cc,covar=TRUE,covar.col=4:5,
copred=obj$coords1,thetaini=c(.1,.2),y.col=3,coords.col=1:2,
fix.nugget=TRUE,nugget=0,kappa=1.2,cutoff=cutoff,trend=~V4+V5,
cov.model=type)

summary(aux2)
```

---

depth *Depths of a geological horizon.*

Description

Dataset previously analyzed by Dubrule and Kostov (1986) and De Oliveira (2005).

Usage

```
data("depth")
```

Format

A data frame with 100 observations on the following 6 variables.

`coord x` x coordinate for depth data.

`coord y` y coordinate for depth data.

`cc` indicator of censure (left and right censure).

`LI` lower limit of censure for depth data.

`LS` upper limit of censure for depth data.

`depth` observated depth.

Details

The observations are placed over a region of about 9 by 5 km and represent depths of a geological horizon measured at 100 locations where 69 points are fully observed and 31 points are censored points, these are divided into left- and right- censored points. The depth data were transformed and their original units remains unknown for confidentiality reasons. For additional details about this dataset we refer to De Oliveira (2005).

References

Dubrule, O. and C. Kostov (1986). An interpolation method taking into account inequality constraints: I. methodology. Mathematical Geology 18(1), 33-51.

De Oliveira, V. (2005). Bayesian inference and prediction of Gaussian random fields based on censored data. Journal of Computational and Graphical Statistics 14(1), 95-115.

Examples

```
data(depth)
summary(depth$depth)
```

`derivcormatrix` *First and second derivates of some correlation matrix*

Description

It computes the matrix of first and second derivates for the exponential, gaussian, matern, spherical, powered exponential and Cauchy correlation matrix.

Usage

```
derivcormatrix(coords, phi, kappa = 0, cov.model = "exponential")
```

Arguments

coords
: 2D spatial coordinates.

phi
: parameter for the matern, powered exponential and cauchy functions.

kappa
: parameter for all correlation functions.

cov.model
: parameter correlation funtion to calculates the derivates in this case 6 functions are avalible "exponential", "gaussian", "matern", "spherical", "powered.exponential", "cauchy".

Details

The correlations functions used to calculate the derivates from this 6 functions are based in the functions by the package geoR (see cov.spatial).

Value

H
: distance matrix.

devR1
: first derivate of the correlation matrix.

devR2
: second derivate of the correlation matrix.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

Diggle, P. & Ribeiro, P. (2007). Model-Based Geostatistics. Springer Series in Statistics.

Gradshtejn, I. S. & Ryzhik, I. M. (1965). Table of integrals, series and products. Academic Press.

See Also

SAEMSCL

Examples

```
n<-200
n1=100
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)

s=derivcormatrix(coords=coords,phi=2,kappa=2,cov.model="exponential")
```

`derivQfun` *Maximum Likelihood Expectation ($logQ$ function and its derivates)*

Description

It computes the $logQ$ function, its derivates of first and second order and the inverse of the hessian matrix for the SAEM estimated parameters.

Usage

```
derivQfun(est, fix.nugget = TRUE)
```

Arguments

`est`	object of the class "SAEMSpatialCens". See `SAEMSCL` function.
`fix.nugget`	(logical) it indicates if the τ^2 parameter must be fixed.

Details

The $logQ$ function refers to the logarithm of the Maximum likelihood conditional expectation, the first and second moments of the truncated normal distribution of censored data are involved in its computation.

Value

`Qlogvalue`	value of the $logQ$ function evaluated in the SAEM estimates.
`gradQ`	gradient for the $logQ$ function evaluated in the SAEM estimates.
`HQ`	hessian Matrix for the $logQ$ function evaluated in the SAEM estimates.
`Qinv`	inverse of the negative Hessian matrix for the $logQ$ function evaluated in the SAEM estimates.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

Diggle, P. & Ribeiro, P. (2007). Model-Based Geostatistics. Springer Series in Statistics.

Gradshtejn, I. S. & Ryzhik, I. M. (1965). Table of integrals, series and products. Academic Press.

See Also

`SAEMSCL`

Examples

```
require(geoR)
data("Missouri")
data=Missouri[1:70,]
data$V3=log((data$V3))



cc=data$V5
y=data$V3
datare1=data
coords=datare1[,1:2]
data1=data.frame(coords,y)
data1=data1[cc==0,]
geodata=as.geodata(data1,y.col=3,coords.col=1:2)
v=variog(geodata)
v1=variofit(v)
cov.ini=c(0,2)

est=SAEMSCL(cc,y,cens.type="left",trend="cte",coords=coords,M=15,perc=0.25,MaxIter=5,pc=0.2,
cov.model="exponential",fix.nugget=TRUE,nugget=2,inits.sigmae=cov.ini[2],inits.phi=cov.ini[1],
search=TRUE,lower=0.00001,upper=50)


d1=derivQfun(est)
d1$QI
```

`distmatrix` *Distance matrix*

Description

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

Usage

```
distmatrix(coords)
```

Arguments

`coords`	2D spatial coordinates.

Value

`dist`	symetric matrix of distances between points.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

Diggle, P. & Ribeiro, P. (2007). Model-Based Geostatistics. Springer Series in Statistics.

See Also

SAEMSCL

Examples

```
n<-200
n1=100

####Simulating spatial coordinates##
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)

H=distmatrix(coords)
```

---

localinfmeas *Local influence measures.*

Description

It computes some measures and plots to asses the local influence of outliers in the SAEM spatial estimation for censored spatial observations, for six types of covariance functions (est$type): "exponential", "matern", "gauss", "spherical","powered.exponential" or "stable" and "cauchy".

Usage

```
localinfmeas(est, fix.nugget = TRUE, diag.plot = TRUE, type.plot = "all", c = 3)
```

Arguments

est	object of the class "SAEMSpatialCens". See SAEMSCL function.
fix.nugget	(logical) it indicates if the τ^2 parameter must be fixed.
diag.plot	(logical) it indicates if diagnostic plots must be showed.
type.plot	type of plot (all: all graphics, rp: response perturbation,smp: scale matrix perturbation, evp: explanatory variable perturbation).
c	constant used for fixing the limit of detection (benchmark value).

Details

this function uses the Maximum likelihood expectation (MLE) under three perturbation schemes, in the response ($M(0)_y$), scale matrix ($M(0)_\Sigma$) and explanatory variables ($M(0)_X$), to detect the influence of outliers in the SAEM estimation procedure.

Value

in addition to the diagnostic graphics (response, scale matrix and explanatory variable schemes, respectively), the function returns the next values.

`Qwrp`	negative Q_{ω_0} matrix under the response perturbation scheme.
`Qwsmp`	negative Q_{ω_0} matrix under the scale matrix perturbation scheme.
`Qwevp`	negative Q_{ω_0} matrix under the explanatory variable perturbation scheme.
`respper`	data.frame containing an indicator of the presence of atypical values and the $M(0)$ values for the response perturbation scheme.
`smper`	data.frame containing an indicator of the presence of atypical values and the $M(0)$ values for the scale matrix perturbation scheme.
`expvper`	a data.frame containing an indicator of the presence of atypical values and the $M(0)$ values for the explanatory variable perturbation scheme.
`limrp`	limit of detection for outliers for the response perturbation scheme.
`limsmp`	limit of detection for outliers for the scale matrix perturbation scheme.
`limevp`	limit of detection for outliers for the explanatory variable perturbation scheme.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

Cook, R. D. (1986). Assessment of local influence. Journal of the Royal Statistical Society, Series B,, 48, 133-169.

Zhu, H., Lee, S., Wei, B. & Zhou, J. (2001). Case-deletion measures for models with incomplete data. Biometrika, 88, 727-737.

See Also

SAEMSCL

Examples

```
require(geoR)
```

```
data("Missouri")
data=Missouri
data$V3=log((data$V3))
cc=data$V5
y=data$V3
n=127
k=1
datare1=data
coords=datare1[,1:2]
data1=data.frame(coords,y)
data1=data1[cc==0,]
geodata=as.geodata(data1,y.col=3,coords.col=1:2)
v=variog(geodata)
v1=variofit(v)
cov.ini=c(0,2)
est=SAEMSCL(cc,y,cens.type="left",trend="cte",coords=coords,M=15,perc=0.25,
MaxIter=5,pc=0.2,cov.model="exponential",fix.nugget=TRUE,nugget=2,
inits.sigmae=cov.ini[2],inits.phi=cov.ini[1], search=TRUE,lower=0.00001,upper=100)


w=localinfmeas(est,fix.nugget=TRUE,c=3)

res=w$respper
res[res[,1]=="atypical obs",]

sm=w$smper
sm[sm[,1]=="atypical obs",]

ev=w$expvper
ev[ev[,1]=="atypical obs",]


##############ANOTHER EXAMPLE#########

n<-200 ### sample size for estimation
n1=100 ### number of observation used in the prediction

###simulated coordinates
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)

coords1=coords[1:n,]

cov.ini=c(0.2,0.1)
type="exponential"
xtot=as.matrix(rep(1,(n+n1)))
xobs=xtot[1:n,]
beta=5

###simulated data
obj=rspacens(cov.pars=c(3,.3,0),beta=beta,x=xtot,coords=coords,cens=0.25,n=(n+n1),
```

```
n1=n1,cov.model=type,cens.type="left")

data2=obj$datare
cc=obj$cc
y=obj$datare[,3]

##### generating atypical observations###
y[91]=y[91]+4
y[126]=y[126]+4
y[162]=y[162]+4
coords=obj$datare[,1:2]

###initial values###
cov.ini=c(0.2,0.1)

est=SAEMSCL(cc,y,cens.type="left",trend="cte",coords=coords,M=15,perc=0.25,
MaxIter=10,pc=0.2,cov.model=type,fix.nugget=TRUE,nugget=0,inits.sigmae=cov.ini[1],
inits.phi=cov.ini[2],search=TRUE,lower=0.00001,upper=50)


w=localinfmeas(est,fix.nugget=TRUE,c=3)

res=w$respper
res[res[,1]=="atypical obs",]

sm=w$smper
sm[sm[,1]=="atypical obs",]

ev=w$expvper
ev[ev[,1]=="atypical obs",]
```

`Missouri` *TCDD concentrations in Missouri (1971).*

Description

Contents the data of TCDD concentrations used for Zirschky et al. in his geostatistical analysis of Hazardous waste data in Missouri.

Usage

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

Format

A data frame with 127 observations on the following 5 variables.

`V1` x coordinate of start of each transect (ft).

`V2` y coordinate of start of each transect (ft).

`V3` TCDD Concentrations (mg/m^3).

`V4` transect length (ft).

`V5` indicator of censure (left censure in all data).

Source

The data was collected in November 1983 by U.S. EPA in several areas of a highway from Missouri. Only the locations used in the geostatistical analysis by the autors are showed.

References

Zirschky, J. H. & Harris, D. J. (1986). Geostatistical analysis of hazardous waste site data. Journal of Environmental Engineering, 112(4), 770-784.

Examples

```
data(Missouri)
summary(Missouri$V3)
```

---

predgraphics — *Prediction graphics for SAEM Algortihm for censored spatial data.*

Description

This function provides prediction raster graphics representation and its standard deviation.

Usage

```
predgraphics(xpred = NULL, grid1, est, points = TRUE,obspoints = 1:sum(est$cc == 0),
colors = terrain.colors(100),sdgraph = TRUE,xlab="X Coord",ylab="Y Coord",
main1="Predicted response", main2="Standard deviation predicted",
xlim=c(min(est$coords[,1]),max(est$coords[,1])),ylim=c(min(est$coords[,2]),
max(est$coords[,2])))
```

Arguments

`xpred`	x design matrix for the prediction coordinates (must be specified when est$trend="other").
`grid1`	grid with the coordinates of the prediction graphics.
`est`	object of class "SAEMSpatialCens".
`points`	(logical), it indicates if some of the observed points may be plotted in the prediction raster graphic (default, points=`TRUE`).
`obspoints`	(vector) if points=`TRUE`, it indicates which of the observed (not censored) values may be plotted in the prediction raster graphics.
`colors`	colors pallete used for the graphics (By default `terrain.colors(100)`).

sdgraph	(logical) it indicates if the standard deviation of the prediction points graphic must be plotted (default sdgraph=TRUE).
xlab	label for x coordinate of the two plots.
ylab	label for y coordinate.
main1	an overall title for the prediction plot.
main2	an overall title for the standard deviation prediction plot.
xlim	x axis limits for the two plots.
ylim	y axis limits for the two plots.

Value

in addition to the raster graphics for prediction, the next values are retorned:

datapred	data.frame with the coordinates and the predicted points used in the prediction raster graphic.
datasdpred	data.frame with the coordinates and the standard deviation predicted points used in the standard deviation prediction raster graphic.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

DELYON, B., LAVIELLE, M.,ANDMOULI NES, E. (1999). Convergence ofa stochastic approximation version of the EM algorithm.Annals of Statistic-s27, 1, 94-128.

Diggle, P. & Ribeiro, P. (2007). Model-Based Geostatistics. Springer Series in Statistics.

See Also

SAEMSCL

Examples

```
data(depth)
cc=depth$cc
y=depth$depth
coords=depth[,1:2]

cov.ini=c(1500,30)
est=SAEMSCL(cc,y,cens.type="left",trend="cte",coords=coords,M=15,perc=0.25,
MaxIter=100,pc=0.2,cov.model="gaussian",fix.nugget=FALSE,nugget=10,
inits.sigmae=cov.ini[2],inits.phi=cov.ini[1], search=TRUE,lower=c(0.00001,0.00001),
upper=c(10000,100))
```

```
coorgra1=seq(min(coords[,1]),max(coords[,1]),length=50)
coorgra2=seq(min(coords[,2]),max(coords[,2]),length=50)

grid1=expand.grid(x=coorgra1,y=coorgra2)
xpred=rep(1,2500)

predgraphics(xpred=xpred,est=est,grid1=grid1,points=TRUE,sdgraph=TRUE)
```

`predSCL` *Prediction for the SAEM algorithm for censored spatial data.*

Description

This function uses the parameters estimates from SAEM to predict values at unknown locations through the MSE criterion assuming normal distribution.

Usage

```
predSCL(xpred, coordspred, est)
```

Arguments

`xpred`	values of the x design matrix for prediction coordinates.
`coordspred`	points coordinates to be predicted.
`est`	object of the class SAEMSpatialCens (see `SAEMSCL` function).

Details

This function predicts using the Mean Square of error (MSE) criterion, that is, it takes the conditional expectation $E(Y|X)$ as the predictor that minimizes the MSE.

Value

`prediction`	prediction value.
`indpred`	indicator for the observed and predicted values (0:observed,1:predicted).
`sdpred`	standard deviation for prediction.
`coordspred`	points coordinates predicted.
`coordsobs`	observed coordinates.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

DELYON, B., LAVIELLE, M.,ANDMOULI NES, E. (1999). Convergence of a stochastic approximation version of the EM algorithm.Annals of Statistic-s27, 1, 94-128.

Diggle, P. & Ribeiro, P. (2007). Model-Based Geostatistics. Springer Series in Statistics.

See Also

SAEMSCL

Examples

```
n<-200 ### sample size for estimation.
n1=100 ### number of observation used in the prediction.

###simulated coordinates
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)

coords=cbind(r1,r2)### coordinates for estimation and prediction.

coords1=coords[1:n,]####coordinates used in estimation.

cov.ini=c(0.2,0.1)###initial values for phi and sigma2.

type="matern"
xtot<-cbind(1,runif((n+n1)),runif((n+n1),2,3))###X matrix for estimation and prediction.

xobs=xtot[1:n,]###X matrix for estimation.
beta=c(5,3,1)

###simulated data
obj=rspacens(cov.pars=c(3,.3,0),beta=beta,x=xtot,coords=coords,kappa=1.2,cens=0.25,
n=(n+n1),n1=n1,cov.model=type,cens.type="left")

data2=obj$datare
cc=obj$cc
y=obj$datare[,3]
coords=obj$datare[,1:2]


#######SAEMSpatialCens object########

est=SAEMSCL(cc,y,cens.type="left",trend="other",x=xobs,coords=coords,kappa=1.2,M=15,
perc=0.25,MaxIter=10,pc=0.2,cov.model="exponential",fix.nugget=TRUE,nugget=0,
inits.sigmae=cov.ini[2],inits.phi=cov.ini[1],search=TRUE,lower=0.00001,upper=50)
```

```
coordspred=obj$coords1
xpred=xtot[(n+1):(n+n1),]
h=predSCL(xpred,coordspred,est)
```

`rspacens` *Censored Spatial data simulation*

Description

It simulates spatial data with linear structure for one type of censure (left or right).

Usage

```
rspacens(cov.pars,beta,x=as.matrix(rep(1,n)),coords,kappa=0,cens,n,n1,
cov.model="exponential",cens.type)
```

Arguments

`cov.pars`	covariance structure parameters for the errors distribution (ϕ, σ^2, τ^2).
`beta`	linear regression parameters.
`x`	design matrix.
`coords`	coordinates of simulated data.
`kappa`	κ parameter used in some covariance structures.
`cens`	percentage of censure in the data (number between 0 and 1).
`n`	number of simulated data used in estimation.
`n1`	number of simulated data used for cross validation (Prediction).
`cov.model`	covariance structure for the data (see `cov.spatial` from `geoR`).
`cens.type`	type of censure (`"left"` or `"right"`).

Details

This function analyses prediction in spatial data. It returns a spatial dataset for estimation (n length) and a spatial dataset (n1 length) used to evaluate the prediction power of a model through cross validation. The covariance functions used here were provided by cov.spatial from the geoR package.

Value

`y`	complete simulated data ($(n + n1)$ length).
`datare`	data frame that will be used for the model estimation (coordinates and response).
`valre`	data that will be used for cross validation studies (just response).
`cc`	indicator of censure (1:censored 0:observed).
`cutoff`	limit of detection simulated for censure (left: <=cutoff, right: > cutoff).
`coords1`	coordinates of value data.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

Diggle, P. & Ribeiro, P. (2007). Model-Based Geostatistics. Springer Series in Statistics.

Schelin, L. & Sjostedt-de Luna, S. (2014). Spatial prediction in the presence of left-censoring. Computational Statistics and Data Analysis, 74.

See Also

SAEMSCL

Examples

```
n<-200 ### sample size for estimation.
n1=100 ### number of observation used in the prediction.

###simulated coordinates
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)### total coordinates (used in estimation and prediction).
coords1=coords[1:n,]####coordinates used for estimation.

type="matern"### covariance structure.

xtot<-cbind(1,runif((n+n1)),runif((n+n1),2,3))## X matrix for estimation and prediction.
xobs=xtot[1:n,]## X matrix for estimation

obj=rspacens(cov.pars=c(3,.3,0),beta=c(5,3,1),x=xtot,coords=coords,
kappa=1.2,cens=0.25,n=(n+n1),n1=n1,cov.model=type,cens.type="left")
```

SAEMSCL *SAEM Algorithm estimation for censored spatial data.*

Description

It estimates the parameters for a linear spatial model with censored observations

Usage

```
SAEMSCL(cc, y, cens.type="left", trend = "cte", LI = NULL, LS = NULL, x = NULL, coords,
kappa = 0, M = 20, perc = 0.25, MaxIter = 300, pc = 0.2, cov.model = "exponential",
fix.nugget = TRUE, nugget, inits.sigmae, inits.phi, search = FALSE, lower, upper)
```

Arguments

`cc` (binary vector) indicator of censure (1: censored observation 0: observed).

`y` (vector) corresponds to response variable.

`cens.type` type of censure (`"left"`:left or `"right"`:right).

`trend` linear trends options: `"cte"`, `"1st"`, `"2nd"` and `"other"`, the three first are defined like in `geoR`, if trend=`"other"`, x (design matrix) must be defined.

`LI` (vector) lower limit, if cens.type=`"both"`, LI must be provided, if cens.type=`"left"` or `"right"` LI and LS are defined by the function through the indicator of censure cc.

`LS` (vector) upper limit, if `cens.type="both"`, LS must be provided, if `cens.type="left"` or "right" LI and LS are defined by the function through the indicator of censure cc.

`x` design matrix.

`coords` corresponds to the coordinates of the spatial data (2D coordinates).

`kappa` value of kappa used in some covariance functions.

`M` number of montecarlo samples for stochastic aproximation.

`perc` percentage of burn-in on the Monte Carlo sample. Default=0.25.

`MaxIter` maximum of iterations for the algorithm.

`pc` percentage of initial iterations of the SAEM algorithm. (Default=0.2).

`cov.model` covariance Structure (see, `cov.spatial` from `geoR`).

`fix.nugget` (logical) indicates if the τ^2 parameter must be fixed.

`nugget` if `fix.nugget=TRUE`, the algorithm just estimates β, σ^2, and ϕ, and fixed τ^2 like nugget, else, τ^2 is estimated and nugget corresponds to initial value for τ^2.

`inits.sigmae` corresponds to initial value for σ^2.

`inits.phi` corresponds to initial value for ϕ parameter.

`search` (logical) this argument gives bounds where the optim routine can find the solution that maximizes the Maximum likelihood expectation. If `search=F`, the optim routine will try to search the solutions for maximization in all the domain for ϕ and τ^2 (if `fix.nugget=FALSE`). If `search=TRUE`, the optim routine search the solutions in a specific neighborhood. We recommended to use `search=F` (see details).

`lower` (vector or numeric) lower bound from the optim solution. If `fix.nugget=T`, lower is numerical and corresponds to the lower bound for search the solution of the ϕ parameter, if `fix.nugget=FALSE` lower is a vector and corresponds to the lower bounds for search the solution of ϕ and $\tau 2$ that maximizes the Maximum Likelihood Expectation (see details).

`upper` (vector or numeric) upper bound from the optim solution. If `fix.nugget==T`, lower is numerical and corresponds to the lower bound for searching the solution of the phi parameter, if `fix.nugget==F`, lower is a vector and corresponds to the lower bounds for searching the solution for ϕ and τ^2 parameters that maximizes the Maximum Likelihood Expectation

Details

The estimation process was computed via SAEM algorithm initially proposed by Deylon et. al.(1999). This is a stochastic approximation of the EM procedure. This procedure circunvent the heavy computational time involved in the MCEM procedure necessary for estimating phi and tau2 parameters (when tau2 is not fixed) since there is not an analytical solution. The search interval was proposed because sometimes the maximization procedure used by optim function does not work for large intervals.

Value

`beta`	estimated β.
`sigma2`	estimated σ^2.
`phi`	estimated ϕ.
`nugget`	estimated or fixed τ^2.
`Theta`	estimated parameters in all iterations (β, σ^2, ϕ) or $(\beta, \sigma^2, \phi, \tau^2)$ if `fix.nugget=F`.
`loglik`	log likelihood for SAEM method.
`AIC`	Akaike information criteria.
`BIC`	Bayesian information criteria.
`AICcorr`	corrected AIC by the number of parameters.
`X`	design matrix.
`Psi`	estimated covariance matrix.
`theta`	final estimation of $\theta = (\beta, \sigma^2, \phi)$ or $\theta = (\beta, \sigma^2, \phi, \tau^2)$ if `fix.nugget=F`.
`uy`	stochastic approximation of the first moment for the truncated normal distribution.
`uyy`	stochastic approximation of the second moment for the truncated normal distribution.
`cc`	indicator of censure (0:observed, 1: censored).
`type`	covariance structure considered in the model.
`kappa`	κ parameter for some covariance structures.
`coords`	coordinates of the observed data.
`iterations`	number of iterations needed to convergence.
`fitted`	fitted values for the SAEM algortihm.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

DELYON, B., LAVIELLE, M.,ANDMOULI NES, E. (1999). Convergence ofa stochastic approximation version of the EM algorithm.Annals of Statistic-s27, 1, 94-128.

Diggle, P. & Ribeiro, P. (2007). Model-Based Geostatistics. Springer Series in Statistics.

See Also

localinfmeas, derivQfun

Examples

```
n<-200 ### sample size for estimation.
n1=100 ### number of observation used in the prediction.

###simulated coordinates
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)

coords1=coords[1:n,]

type="matern"
#xtot<-cbind(1,runif((n+n1)),runif((n+n1),2,3))
xtot=as.matrix(rep(1,(n+n1)))
xobs=xtot[1:n,]
beta=5
#beta=c(5,3,1)

###simulated data
obj=rspacens(cov.pars=c(3,.3,0),beta=beta,x=xtot,coords=coords,kappa=1.2,
cens=0.25,n=(n+n1),n1=n1,cov.model=type,cens.type="left")

data2=obj$datare
cc=obj$cc
y=obj$datare[,3]
coords=obj$datare[,1:2]
##initials values obtained from variofit.
cov.ini=c(0.13,0.86)

est=SAEMSCL(cc,y,cens.type="left",trend="cte",coords=coords,
kappa=1.2,M=15,perc=0.25,MaxIter=10,pc=0.2,cov.model=type,
fix.nugget=TRUE,nugget=0,inits.sigmae=cov.ini[1],
inits.phi=cov.ini[2],search=TRUE,lower=0.00001,upper=100)

summary(est)
```

Seminaive *Seminaive algorithm for spatial censored prediction.*

Description

This function executes the seminaive algorithm proposed by Schelin et al. (2014)

Usage

```
Seminaive(data, y.col, coords.col, covar, covar.col, copred,cov.model = "exponential",
thetaini, fix.nugget = TRUE, nugget,kappa = 0, cons, MaxIter, cc, cutoff, trend)
```

Arguments

`data`	data.frame containing the coordinates, covariates and response variable.
`y.col`	(numeric) column of data.frame that corresponds to the response variable.
`coords.col`	(numeric) columns of data.frame that corresponds to the coordinates of the spatial data.
`covar`	(logical) indicates the presence of covariates in the spatial censored estimation (`FALSE`: without covariates, `TRUE`: with covariates).
`covar.col`	(numeric) columns of data.frame that corresponds to the covariates in the spatial censored linear model estimation.
`copred`	coordinates used in the prediction procedure.
`cov.model`	covariance model in the structure of covariance (see `cov.spatial` from `geoR`).
`thetaini`	initial values for the σ^2 and ϕ values in the covariance structure.
`fix.nugget`	(logical) it indicates if the τ^2 parameter must be fixed.
`nugget`	(numeric) values of the τ^2 parameter, if `fix.nugget=F`, this value corresponds to an initial value.
`kappa`	value of κ involved in some covariance functions.
`cons`	(vector) vector containing the (c_1, c_2, c_3) constants used in the convergence criterion for the algorithm (see Schedlin).
`MaxIter`	maximum of iterations for the algorithm.
`cc`	(binary vector) indicator of censure (1: censored, 0: observed)
`cutoff`	(vector) limit of detection for censure (rc: >cutoff, lc: <cutoff)
`trend`	it specifies the mean part of the model. See documentation of `trend.spatial` from `geoR` for further details. By default `"cte"`.

Details

This function estimates and computes predictions following Schedlin et al. (2014). See reference.

Value

`zk`	vector with observed and estimate censored observations by kriging prediction.
`AIC`	AIC of the estimated model.
`BIC`	BIC of the estimated model.
`beta`	beta parameter for the mean structure.

`theta`	vector of estimate parameters for the mean and covariance structure $(\beta, \sigma^2, \phi, \tau^2)$.
`predictions`	Predictions obtained for the seminaive algorithm.
`sdpred`	Standard deviations of predictions.
`loglik`	log likelihood from the estimated model.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

Schelin, L. & Sjostedt-de Luna, S. (2014). Spatial prediction in the presence of left-censoring. Computational Statistics and Data Analysis, 74.

See Also

SAEMSCL

Examples

```
n<-200 ### sample size for estimation.
n1=100 ### number of observation used in the prediction.

###simulated coordinates.
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)### total coordinates (used in estimation and prediction).
coords1=coords[1:n,]####coordinates used for estimation.

type="matern"### covariance structure.

xtot<-cbind(1,runif((n+n1)),runif((n+n1),2,3))## X matrix for estimation and prediction.
xobs=xtot[1:n,]## X matrix for estimation.

###simulated data.
obj=rspacens(cov.pars=c(3,.3,0),beta=c(5,3,1),x=xtot,coords=coords,kappa=1.2,
cens=0.25,n=(n+n1),n1=n1,cov.model=type,cens.type="left")

data2=obj$datare
data2[,4:5]=xobs[,-1]

cc=obj$cc
y=obj$datare[,3]
cutoff=rep(obj$cutoff,length(y[cc==1]))

###seminaive algorithm
```

```
r=Seminaive(data=data2,y.col=3,covar=TRUE,coords.col=1:2,covar.col=4:5,cov.model="matern",
thetaini=c(.1,.2),fix.nugget=TRUE,nugget=0,kappa=1.5,cons=c(0.1,2,0.5),MaxIter=100,
cc=obj$cc,cutoff=cutoff,copred=obj$coords1,trend=~V4+V5)

summary(r)
```

---

summary.naive *Summary of a naive object*

Description

summary method for class "naive".

Usage

```
## S3 method for class 'naive'
summary(object,...)
```

Arguments

object	object of the class "naive" (see algnaive12 function).
...	Additional arguments.

Value

mean.str1	Estimates for the mean structure parameters **beta** for Naive 1 method.
var.str1	Estimates for the variance structure parameters σ^2, ϕ for Naive 1 method.
mean.str2	Estimates for the mean structure parameters **beta** for Naive 2 method.
var.str2	Estimates for the variance structure parameters σ^2, ϕ for Naive 2 method.
predictions1	predictions for Naive 1 method.
predictions2	predictions for Naive 1 method.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

Schelin, L. & Sjostedt-de Luna, S. (2014). Spatial prediction in the presence of left-censoring. Computational Statistics and Data Analysis, 74.

See Also

SAEMSCL

Examples

```
n<-200 ### sample size for estimation.
n1=100 ### number of observation used for prediction.

###simulated coordinates
n<-200 ### sample size for estimation.
n1=100 ### number of observation used in the prediction.

###simulated coordinates
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)### total coordinates (used in estimation and prediction).

coords1=coords[1:n,]####coordinates used for estimation.

type="matern"### covariance structure.

xtot<-cbind(1,runif((n+n1)),runif((n+n1),2,3))## X matrix for estimation and prediction.
xobs=xtot[1:n,]## X matrix for estimation.

###simulated data
obj=rspacens(cov.pars=c(3,.3,0),beta=c(5,3,1),x=xtot,coords=coords,kappa=1.2,
cens=0.25,n=(n+n1),n1=n1,cov.model=type,cens.type="left")

data2=obj$datare
data2[,4:5]=xobs[,-1]

cc=obj$cc
y=obj$datare[,3]
cutoff=rep(obj$cutoff,length(y[cc==1]))


aux2=algnaive12(data=data2,cc=obj$cc,covar=TRUE,covar.col=4:5,
copred=obj$coords1,thetaini=c(.1,.2),y.col=3,coords.col=1:2,
fix.nugget=TRUE,nugget=0,kappa=1.2,cutoff=cutoff,trend=~V4+V5,
cov.model=type)

summary(aux2)
```

---

summary.SAEMSpatialCens *Summary of a SAEMSpatialCens object.*

Description

summary method for class "SAEMSpatialCens".

Usage

```
## S3 method for class 'SAEMSpatialCens'
summary(object,...)
```

Arguments

`object`	object of the class "SAEMSpatialCens" (see `SAEMSCL` function).
`...`	Additional arguments.

Value

`mean.str`	Estimates for the mean structure parameters **beta** for SAEMSCL method.
`var.str`	Estimates for the variance structure parameters σ^2, ϕ for SAEMSCL method.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

DELYON, B., LAVIELLE, M.,ANDMOULI NES, E. (1999). Convergence of a stochastic approximation version of the EM algorithm.Annals of Statistic-s27, 1, 94-128.

Diggle, P. & Ribeiro, P. (2007). Model-Based Geostatistics. Springer Series in Statistics.

See Also

`SAEMSCL`

Examples

```
n<-200 ### sample size for estimation.
n1=50 ### number of observation used in the prediction.

###simulated coordinates
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)

coords1=coords[1:n,]

type="matern"
#xtot<-cbind(1,runif((n+n1)),runif((n+n1),2,3))
xtot=as.matrix(rep(1,(n+n1)))
xobs=xtot[1:n,]
```

```
beta=5
#beta=c(5,3,1)

###simulated data
obj=rspacens(cov.pars=c(3,.3,0),beta=beta,x=xtot,coords=coords,kappa=1.2,
cens=0.25,n=(n+n1),n1=n1,cov.model=type,cens.type="left")

data2=obj$datare
cc=obj$cc
y=obj$datare[,3]
coords=obj$datare[,1:2]

est=SAEMSCL(cc,y,cens.type="left",trend="cte",coords=coords,
kappa=1.2,M=15,perc=0.25,MaxIter=10,pc=0.2,cov.model=type,
fix.nugget=TRUE,nugget=0,inits.sigmae=cov.ini[1],
inits.phi=cov.ini[2],search=TRUE,lower=0.00001,upper=100)

summary(est)
```

`summary.seminaive` *Summary of a seminaive object*

Description

`summary` method for class "seminaive".

Usage

```
## S3 method for class 'seminaive'
summary(object,...)
```

Arguments

`object`	object of the class "seminaive" (see `Seminaive` function).
`...`	Additional arguments.

Value

`mean.str`	Estimates for the mean structure parameters **beta** for seminaive method.
`var.str`	Estimates for the variance structure parameters σ^2, ϕ for seminaive method.
`predictions`	predictions for seminaive method.

Author(s)

Alejandro Ordonez «ordonezjosealejandro@gmail.com», Victor H. Lachos «hlachos@ime.unicamp.br» and Christian E. Galarza «cgalarza88@gmail.com»

Maintainer: Alejandro Ordonez «ordonezjosealejandro@gmail.com»

References

Schelin, L. & Sjostedt-de Luna, S. (2014). Spatial prediction in the presence of left-censoring. Computational Statistics and Data Analysis, 74.

See Also

SAEMSCL

Examples

```
n<-200 ### sample size for estimation.
n1=100 ### number of observation used in the prediction.

###simulated coordinates.
r1=sample(seq(1,30,length=400),n+n1)
r2=sample(seq(1,30,length=400),n+n1)
coords=cbind(r1,r2)### total coordinates (used in estimation and prediction).
coords1=coords[1:n,]####coordinates used for estimation.

type="matern"### covariance structure.

xtot<-cbind(1,runif((n+n1)),runif((n+n1),2,3))## X matrix for estimation and prediction.
xobs=xtot[1:n,]## X matrix for estimation.

###simulated data.
obj=rspacens(cov.pars=c(3,.3,0),beta=c(5,3,1),x=xtot,coords=coords,kappa=1.2,
cens=0.25,n=(n+n1),n1=n1,cov.model=type,cens.type="left")

data2=obj$datare
data2[,4:5]=xobs[,-1]

cc=obj$cc
y=obj$datare[,3]
cutoff=rep(obj$cutoff,length(y[cc==1]))

###seminaive algorithm
r=Seminaive(data=data2,y.col=3,covar=TRUE,coords.col=1:2,covar.col=4:5,cov.model="matern",
thetaini=c(.1,.2),fix.nugget=TRUE,nugget=0,kappa=1.5,cons=c(0.1,2,0.5),MaxIter=100,
cc=obj$cc,cutoff=cutoff,copred=obj$coords1,trend=~V4+V5)

summary(r)
```

Index