Package 'lgcpSPDE'

January 20, 2017

Type Package

Title Fitting and Simulation for log-Gaussian Cox processes using INLA/SPDE approach

Version 0.0.1

Date 2016-05-05

Author Charlotte Jones-Todd <cmjt@st-andrews.ac.uk>

Maintainer Charlotte Jones-Todd <cmjt@st-andrews.ac.uk>

Depends R (>= 3.0.0), INLA, spatstat, maptools, fields, TMB, Rcpp

Imports Matrix

LinkingTo Rcpp, TMB, Matrix

Description This package provides functions for the simulation and parameter estimation of log Gaussian Cox point process models using INLA/SPDE approach.

License GPL

URL https://github.com/cmjt/lgcpSPDE

LazyData TRUE

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

compile.lgcpSPDE *Compiling TMB C++ templates*

Description

Compiles the lgcpSPDE TMB templates into a shared object file. This must be done a single time following installation or updating of the package.

Usage

```
compile.lgcpSPDE()
```

find.fields *Function that extracts the "random fields" of the model fitted.*

Description

Plots the estimated random fields or parameter posterior densities from an object returned by mark.pp.fit().

Usage

```
find.fields(x = NULL, mesh = NULL, n.t = NULL, sd = FALSE,
  plot = FALSE, spatial.polygon = NULL, ...)
```

Arguments

x	A fitted model from mark.pp.fit().
mesh	the mesh used in the model fit.
n.t	numeric, the number of time points.
sd	Logical, if FALSE means of random fields aer returned.
plot	Logical, if TRUE the returned matricies (either SD or Mean of random fields are plotted.
spatial.polygon	Optional, if a spatial polygon of the domain is supplied, only values of the random field within the domain will be returned
...	additional graphical parameters @importFrom fields image.plot

`fit.lgcp` *Function to fit a spatio-temporal log-Gaussian Cox process model with an intercept and covariates (optional)*

Description

Function to fit a spatio-temporal log-Gaussian Cox process model with an intercept and covariates (optional)

Usage

```
fit.lgcp(mesh = NULL, mesh.pars = NULL, locs = NULL, temp = NULL,
  covariates = NULL, prior.rho = list(theta = list(prior = "pccor1", param =
  c(0, 0.9))), verbose = FALSE, control.inla = list(strategy = "gaussian",
  int.strategy = "eb"), return.attributes = FALSE, ns = NULL)
```

Arguments

`mesh`	a "mesh" object i.e. delauney triangulation of the domain, an object returned by make.mesh.
`mesh.pars`	a named vertor of mesh parameters, must contain `cutoff` length at which to cut off triangle edge lengths, `min` triangle edge length inside region, and `max` triangle edge length outside region.
`locs`	a matrix of observation locations, where each row corresponds to the observation.
`temp`	a numeric vector specifying a temporal index for each observation (starting at 1.....T).
`covariates`	a named data.frame of covariates
`prior.rho`	prior for the temporal correlation coefficient, by default a `pcprior` is used with `param=c(0-0.9)`.
`verbose`	Logical if `TRUE` model fit is output to screen.
`response`	a vector of response variable, each corresponds to the spatial locations in `locs`.

Value

A `inla` result object

fit.marked.lgcp *Fitting a marked point process model*

Description

Fits a spatio-temporal marked point process model using INLA coupled with the SPDE approch

Usage

```
fit.marked.lgcp(mesh = NULL, locs = NULL, t.index = NULL, mark = NULL,
  covariates = NULL, mark.family = "gaussian", verbose = FALSE,
  prior.rho = list(theta = list(prior = "pccor1", param = c(0, 0.9))),
  hyper = list(theta = list(prior = "normal", param = c(0, 10))))
```

Arguments

mesh	delauney triangulation of area, an object returned by make.mesh is suitable.
locs	a matrix of nrow locations in ncol dimesions.
t.index	a vector of length nrow of time index units refering to each point location given in locs.
mark	a vector of length nrow of marks refering to each point location
mark.family	assumed likelihood for mark, by defalt "gaussian".
verbose	Logical, if TRUE, model fitting is output the console.

Value

An inla model fit object,

geo.fit *Function to fit a spatial geo-statistical "prediction' model*

Description

Function to fit a spatial geo-statistical "prediction' model

Usage

```
geo.fit(mesh = NULL, locs.o = NULL, locs.p = NULL, response = NULL,
  family = "gaussian", verbose = FALSE)
```

Arguments

mesh	a "mesh" object i.e. delauney triangulation of the domain, an object returned by make.mesh.
locs.o	a matrix of observation locations, where each row corresponds to the observation.
locs.p	a matrix of observation locations, where each row corresponds to the prediction location.
response	a vector of response variables, each corresponds to the spatial locations in locs.
family	a character specifying the assumed likelihood of the response, by default is Gaussian.
verbose	Logical if TRUE model fit is output to screen.

Value

A inla result object

---

geo.joint.fit *Function to fit a joint spatio-temporal model to geo-statistical data where one spatio- temporal component is shared between the responses*

Description

Function to fit a joint spatio-temporal model to geo-statistical data where one spatio- temporal component is shared between the responses

Usage

```
geo.joint.fit(mesh = NULL, locs.1 = NULL, locs.2 = NULL,
  response.1 = NULL, response.2 = NULL, t.index = NULL,
  covariates = NULL, family = c("gaussian", "gaussian"), verbose = FALSE,
  prior.rho = list(theta = list(prior = "pccor1", param = c(0, 0.9))),
  hyper = list(theta = list(prior = "normal", param = c(0, 10))),
  control.inla = list(strategy = "gaussian", int.strategy = "eb"))
```

Arguments

mesh	a "mesh" object i.e. delauney triangulation of the domain, an object returned by make.mesh.
covariates	a named data.frame of covariates
family	a character vector specifying the assumed likelihoods of the response, by default is c("gaussian,"gaussian").
verbose	Logical if TRUE model fit is output to screen.

prior.rho	prior for the temporal correlation coefficient, by default a pcprior is used with param=c(0-0.9).
locs	a matrix of observation locations, where each row corresponds to the observation.
response	a matrix of response variables, each row corresponds to the spatial locations in locs, the first column is the first response as a result of a spatio-temporal process we want to "copy" into the second column response.
temp	a numeric vector specifying a temporal index for each observation (starting at 1.....T)

Value

A inla result object

---

geo.st.fit *Function to fit a spatio-temporal model to geo-statistical data with an intercept and covariates*

Description

Function to fit a spatio-temporal model to geo-statistical data with an intercept and covariates

Usage

```
geo.st.fit(mesh = NULL, locs = NULL, response = NULL, temp = NULL,
  covariates = NULL, family = "gaussian", prior.rho = list(theta =
  list(prior = "pccor1", param = c(0, 0.9))), verbose = FALSE)
```

Arguments

mesh	a "mesh" object i.e. delauney triangulation of the domain, an object returned by make.mesh.
locs	a matrix of observation locations, where each row corresponds to the observation.
response	a vector of response variable, each corresponds to the spatial locations in locs.
temp	a numeric vector specifying a temporal index for each observation (starting at 1.....T).
covariates	a named data.frame of covariates
family	a character vector specifying the assumed likelihood of the response, by default is "gaussian".
prior.rho	prior for the temporal correlation coefficient, by default a pcprior is used with param=c(0-0.9).
verbose	Logical if TRUE model fit is output to screen.

Value

A inla result object

make.mesh *Wrapper function to create a mesh object as used in an INLA model fit*

Description

Wrapper function to create a mesh object as used in an INLA model fit

Usage

```
make.mesh(locs = NULL, mesh.pars = NULL, spatial.polygon = NULL,
  sphere = FALSE, plot = FALSE)
```

Arguments

locs	a matrix of locations, either this or spatial.polygon must be supplied.
mesh.pars	a named vertor of mesh parameters, must contain cutoff length at which to cut off triangle edge lengths, min triangle edge length inside region, and max triangle edge length inside region.
spatial.polygon	if supplied the spatial polygon for the domain is used to construct mesh
sphere	Logical if TRUE the mesh is constructed on the unit sphere, note this is only possible if coordinates are longitude and Latitude, by default FALSE
plot	Logical if TRUE the triangulation is plotted

Value

A "mesh" object used in an INLA/SPDE model

rgeospde *Function to simulate from a spatial/spatio temporal SPDE model*

Description

Function to simulate from a spatial/spatio temporal SPDE model

Usage

```
rgeospde(locs = NULL, mesh = NULL, kappa = NULL, sigma2 = 1, n = 1,
  rho = 0.9, seed = 1, non.stat = NULL)
```

Arguments

locs	a matrix of locations at which the values are to be simulated
mesh	a "mesh" object i.e. delauney triangulation of the domain, an object returned by make.mesh.
kappa	a numeric constant, parameter of the SPDE model.
sigma2	a numeric constant, parameter of the SPDE model, by default this is 1.
n	a numeric constant defining the number of time points, by default 1.
rho	the ar1 correlation coefficient for spatio-temporal samples, by default this is 0.9.
seed	seed for the simulation, by default this is 1
non.stat	a named list the first element fn the spatial function which kappa varies with and theta a vctor of length 3 specifying the theta values of the non-stationary model

Value

A matrix of values each column a set of observations at each time point

---

rlgcpspde *Function to simulate from a spatial/spatio temporal LGCP (SPDE model)*

Description

Function to simulate from a spatial/spatio temporal LGCP (SPDE model)

Usage

```
rlgcpspde(spatial.polygon = NULL, mesh.pars = NULL, mu = 0,
  kappa = NULL, sigma2 = 0.05, n = 1, rho = 0.9, mark = FALSE,
  beta = NULL, mark.function = function(x, y) cos(x) - sin(y), seed = 1,
  non.stat = NULL)
```

Arguments

spatial.polygon	the spatial polygon for the domain is used to construct the delauney traingulation
mesh.pars	a named vertor of mesh parameters, must contain cutoff length at which to cut off triangle edge lengths, min triangle edge length inside region, and max triangle edge length inside region.
mu	numeric or a named list, the intercept term to simulate a LGCP, by default is 0. if a named list must contain elements mean, cov.model, cov.pars the latter two parameters of the grf function geoR. to simulate a non-stationary expectation
kappa	a numeric constant, parameter of the SPDE model.
sigma2	a numeric constant, parameter of the SPDE model, by default this is 0.05.

`n`	a numeric constant defining the number of time points, by default 1.
`rho`	the ar1 correlation coefficient for spatio-temporal samples, by default this is 0.9.
`mark`	Logical, if TRUE a marked point pattern is simulated
`beta`	a scalar, this, the interaction parameter describing the dependance between the mark and point locations
`mark.function`	a function of 2D spatial coordinates which describes the spatial process specific to the mark, by default this is `function(x,y) cos(x) - sin(y)`.
`seed`	seed for the simulation, by default this is 1
`non.stat`	a named list the first element `fn` the spatial function which kappa varies with and `theta` a vctor of length 3 specifying the theta values of the non-stationary model

Value

A named matrix (or a list of matricies if spatio-temporal) of point locations and (if a marked point pattern is simulated) a mark values

Index