Package ‘spatsurv’

July 10, 2015

Type Package

Title Bayesian Spatial Survival Analysis with Parametric Proportional Hazards Models

Version 0.9-10

Date 2015-06-24

Author Benjamin M. Taylor and Barry S. Rowlingson

Maintainer Benjamin M. Taylor <b.taylor1@lancaster.ac.uk>

Description Bayesian inference for parametric proportional hazards spatial survival models; flexible spatial survival models.

License GPL-3

Imports survival, sp, spatstat, raster, iterators, RandomFields, fields, rgl, Matrix, stringr, rgeos, RColorBrewer, geostatsp, OpenStreetMap, methods

Suggests rgdal, gpclib

NeedsCompilation no

Repository CRAN

Date/Publication 2015-07-10 20:41:43

R topics documented:

`spatsurv-package` *spatsurv*

Description

An R package for spatially correlated parametric proportional hazards survial analysis.

Usage

```
spatsurv
```

Format

`logi NA`

Details

Package:	spatsurv
Version:	0.9-10
Date:	2015-06-24
License:	GPL-3

sectionDependencies The package `spatsurv` depends upon some other important contributions to CRAN in order to operate; their uses here are indicated:

survival, sp, spatstat, raster, iterators, RandomFields, fields, rgl, Matrix, stringr, RColorBrewer, geostatsp, rgeos.

sectionCitation To cite use of `spatsurv`, the user may refer to the following work:

spatsurv: an `R` Package for Bayesian Inference with Spatial Survival Models.
Benjamin M. Taylor and Barry S. Rowlingson.
Submitted to The Journal Of Statistical Software.

references X

Author(s)

Benjamin Taylor, Health and Medicine, Lancaster University, Barry Rowlingson, Health and Medicine, Lancaster University

---

`allocate` *allocate function*

Description

A function to allocate coordinates to an observation whose spatial location is known to the regional level

Usage

```
allocate(poly, popden, survdat, pid, sid, n = 2, wid = 2000)
```

Arguments

`poly`	a SpatialPolygonsDataFrame, on which the survival data exist in aggregate form
`popden`	a sub-polygon raster image of population density
`survdat`	data.frame containing the survival data

`pid` name of the variable in the survival data that gives the region identifier in poly

`sid` the name of the variable in poly to match the region identifier in survdat to

`n` the number of different allocations to make. e.g. if n is 2 (the default) two candidate sets of locations are available.

`wid` The default is 2000, interpreted in metres ie 2Km. size of buffer to add to window for raster cropping purposes: this ensures that for each polygon, the cropped raster covers it completely.

Value

matrices x and y, both of size (number of observations in survdat x n) giving n potential candidate locations of points in the columns of x and y.

`alpha` *alpha function*

Description

A function used in calculating the coefficients of a B-spline curve

Usage

```
alpha(i, j, knots, knotidx)
```

Arguments

`i` index i

`j` index j

`knots` knot vector

`knotidx` knot index

Value

a vector

`B` *B function*

Description

A recursive function used in calculating the coefficients of a B-spline curve

Usage

```
B(x, i, j, knots)
```

Arguments

`x`	locations at which to evaluate the B-spline
`i`	index i
`j`	index j
`knots`	a knot vector

Value

a vector of polynomial coefficients

`basehazard` *basehazard function*

Description

Generic function for computing the baseline hazard

Usage

```
basehazard(obj, ...)
```

Arguments

`obj`	an object
`...`	additional arguments – currently there are none, but this is for extensibility

Value

method basehazard

See Also

basehazard.basehazardspec, exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

basehazard.basehazardspec
basehazard.basehazardspec function

Description

A function to retrieve the baseline hazard function

Usage

```
## S3 method for class 'basehazardspec'
basehazard(obj, ...)
```

Arguments

obj	an object of class basehazardspec
...	additional arguments – currently there are none, but this is for extensibility

Value

a function returning the baseline hazard

See Also

exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

baselinehazard *baselinehazard function*

Description

A function to compute quantiles of the posterior baseline hazard or cumulative baseline hazard.

Usage

```
baselinehazard(x, t = NULL, n = 100, probs = c(0.025, 0.5, 0.975),
  cumulative = FALSE, plot = TRUE, bw = FALSE, ...)
```

Arguments

`x`	an object inheriting class mcmcspatsurv
`t`	optional vector of times at which to compute the quantiles, Defult is NULL, in which case a uniformly spaced vector of length n from 0 to the maximum time is used
`n`	the number of points at which to compute the quantiles if t is NULL
`probs`	vector of probabilities
`cumulative`	logical, whether to return the baseline hazard (default i.e. FALSE) or cumulative baseline hazard
`plot`	whether to plot the result
`bw`	Logical. Plot in black/white/greyscale? Default is to produce a colour plot. Useful for producing plots for journals that do not accept colour plots.
`...`	additional arguments to be passed to plot

Value

the vector of times and quantiles of the baseline or cumulative baseline hazard at those times

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

---

`betapriorGauss` *betapriorGauss function*

Description

A function to define Gaussian priors for beta. This function simply stores a vector of means and standard deviations to be passed to the main MCMC function, survspat.

Usage

```
betapriorGauss(mean, sd)
```

Arguments

`mean`	the prior mean, a vector of length 1 or more. 1 implies a common mean.
`sd`	the prior standard deviation, a vector of length 1 or more. 1 implies a common standard deviation.

Value

an object of class "betapriorGauss"

See Also

survspat, betapriorGauss, omegapriorGauss, etapriorGauss, indepGaussianprior, derivindepGaussianprior

---

`Bspline.construct` *Bspline.construct function*

Description

A function to construct a B-spline basis matrix for given data and basis coefficients. Used in evaluating the baseline hazard.

Usage

```
Bspline.construct(x, basis)
```

Arguments

`x`	a vector, the data
`basis`	an object created by the getBbasis function

Value

a basis matrix

`BsplineHaz` *BsplineHaz function*

Description

A function to define a parametric proportional hazards model where the baseline hazard is modelled by a basis spline. This function returns an object inheriting class 'basehazardspec', list of functions 'distinfo', 'basehazard', 'gradbasehazard', 'hessbasehazard', 'cumbasehazard', 'gradcumbasehazard', 'hesscumbasehazard' and 'densityquantile'

Usage

```
BsplineHaz(times, knots = quantile(times), degree = 3, MLinits = NULL)
```

Arguments

times	vector of survival times (both censored and uncensored)
knots	vector of knots in ascending order, must include minimum and maximum values of 'times'
degree	degree of the spline basis, default is 3
MLinits	optional starting values for the non-spatial maximisation routine using optim. Note that we are working with the log of the parameters. Default is -10 for each parameter.

Details

The distinfo function is used to provide basic distribution specific information to other spatsurv functions. The user is required to provide the following information in the returned list: npars, the number of parameters in this distribution; parnames, the names of the parameters; trans, the transformation scale on which the priors will be provided; itrans, the inverse transformation function that will be applied to the parameters before the hazard, and other functions are evaluated; jacobian, the derivative of the inverse transformation function with respect to each of the parameters; and hessian, the second derivatives of the inverse transformation function with respect to each of the parameters – note that currently the package spatsurv only allows the use of functions where the parameters are transformed independently.

The basehazard function is used to evaluate the baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, t and returns a vector.

The gradbasehazard function is used to evaluate the gradient of the baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, t, and returns a matrix.

The hessbasehazard function is used to evaluate the Hessian of the baseline hazard function. It returns a function that accepts as input a vector of times, t and returns a list of hessian matrices corresponding to each t.

The cumbasehazard function is used to evaluate the cumulative baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, t and returns a vector.

The gradcumbasehazard function is used to evaluate the gradient of the cumulative baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, t, and returns a matrix.

The hesscumbasehazard function is used to evaluate the Hessian of the cumulative baseline hazard function. It returns a function that accepts as input a vector of times, t and returns a list of hessian matrices corresponding to each t.

The densityquantile function is used to return quantiles of the density function. This is NOT REQUIRED for running the MCMC, merely for us in post-processing with the predict function where type is 'densityquantile'. In the case of the Weibull model for the baseline hazard, it can be shown that the q-th quantile is:

Value

an object inheriting class 'basehazardspec'

See Also

exponentialHaz, gompertzHaz, makehamHaz, weibullHaz

`checkSurvivalData` *checkSurvivalData function*

Description

A function to check whether the survival data to be passed to survspat is in the correct format

Usage

```
checkSurvivalData(s)
```

Arguments

`s`	an object of class Surv, from the survival package

Value

if there are any issues with data format, these are returned with the data an error message explaining any issues with the data

---

`circulant` *circulant function*

Description

generic function for constructing circulant matrices

Usage

```
circulant(x, ...)
```

Arguments

`x`	an object
`...`	additional arguments

Value

method circulant

circulant.matrix *circulant.matrix function*

Description

If x is a matrix whose columns are the bases of the sub-blocks of a block circulant matrix, then this function returns the block circulant matrix of interest.

Usage

```
## S3 method for class 'matrix'
circulant(x, ...)
```

Arguments

`x`	a matrix object
`...`	additional arguments

Value

If x is a matrix whose columns are the bases of the sub-blocks of a block circulant matrix, then this function returns the block circulant matrix of interest.

circulant.numeric *circulant.numeric function*

Description

returns a circulant matrix with base x

Usage

```
## S3 method for class 'numeric'
circulant(x, ...)
```

Arguments

`x`	an numeric object
`...`	additional arguments

Value

a circulant matrix with base x

`circulantij` *circulantij function*

Description

A function to return the "idx" i.e. c(i,j) element of a circulant matrix with base "base".

Usage

```
circulantij(idx, base)
```

Arguments

`idx`	vector of length 2 th (i,j) (row,column) index to return
`base`	the base matrix of a circulant matrix

Value

the ij element of the full circulant

`covmodel` *covmodel function*

Description

A function to define the spatial covariance model, see also ?CovarianceFct. Note that the parameters defined by the 'pars' argument are fixed, i.e. not estimated by the MCMC algorithm. To have spatsurv estimate these parameters, the user must construct a new covariance function to do so, see the spatsurv vignette.

Usage

```
covmodel(model, pars)
```

Arguments

`model`	correlation type, a string see ?CovarianceFct
`pars`	vector of additional parameters for certain classes of covariance function (eg Matern), these must be supplied in the order given in ?CovarianceFct and are not estimated

Value

an object of class covmodel

See Also

CovarianceFct

CSplot *CSplot function*

Description

A function to produce a diagnostic plot for model fit using the Cox-Snell residuals.

Usage

```
CSplot(mod, plot = TRUE, bw = FALSE, ...)
```

Arguments

mod	an object produced by the function survspat
plot	whether to plot the result, default is TRUE
bw	Logical. Plot in black/white/greyscale? Default is to produce a colour plot. Useful for producing plots for journals that do not accept colour plots.
...	other arguments to pass to plot

Value

the x and y values used in the plot

cumbasehazard *cumbasehazard function*

Description

Generic function for computing the cumulative baseline hazard

Usage

```
cumbasehazard(obj, ...)
```

Arguments

obj	an object
...	additional arguments – currently there are none, but this is for extensibility

Value

method cumbasehazard

See Also

cumbasehazard.basehazardspec, exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

`cumbasehazard.basehazardspec`
cumbasehazard.basehazardspec function

Description

A function to retrieve the cumulative baseline hazard function

Usage

```
## S3 method for class 'basehazardspec'
cumbasehazard(obj, ...)
```

Arguments

`obj`	an object of class basehazardspec
`...`	additional arguments – currently there are none, but this is for extensibility

Value

a function returning the cumulative baseline hazard

See Also

exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

`cumulativeBspline.construct`
cumulativeBspline.construct function

Description

A function to construct the integral of a B-spline curve given data and basis coefficients. Used in evaluating the cumulative baseline hazard.

Usage

```
cumulativeBspline.construct(x, basis)
```

Arguments

`x`	a vector, the data
`basis`	an object created by the getBbasis function

Value

an object that allows the integral of a given B-spline curve to be computed

densityquantile   *densityquantile function*

Description

Generic function for computing quantiles of the density function for a given baseline hazard. This may not be analytically tractable.

Usage

```
densityquantile(obj, ...)
```

Arguments

`obj`	an object
`...`	additional arguments – currently there are none, but this is for extensibility

Value

method densityquantile

See Also

densityquantile.basehazardspec, exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

densityquantile.basehazardspec   *densityquantile.basehazardspec function*

Description

A function to retrieve the quantiles of the density function

Usage

```
## S3 method for class 'basehazardspec'
densityquantile(obj, ...)
```

Arguments

`obj`	an object of class basehazardspec
`...`	additional arguments – currently there are none, but this is for extensibility

Value

a function returning the density quantiles

See Also

exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

---

densityquantile_PP *densityquantile_PP function*

Description

A function to compute quantiles of the density function

Usage

```
densityquantile_PP(inputs)
```

Arguments

inputs	inputs for the function including the model matrix, frailties, fixed effects and the parameters of the baseline hazard derived from this model

Value

quantiles of the density function for the individual

---

density_PP *density_PP function*

Description

A function to compute an individual's density function

Usage

```
density_PP(inputs)
```

Arguments

inputs	inputs for the function including the model matrix, frailties, fixed effects and the parameters of the baseline hazard derived from this model

Value

the density function for the individual

derivindepGaussianprior

derivindepGaussianprior function

Description

A function for evaluating the first and second derivatives of the log of an independent Gaussian prior

Usage

```
derivindepGaussianprior(beta = NULL, omega = NULL, eta = NULL, priors)
```

Arguments

`beta`	a vector, the parameter beta
`omega`	a vector, the parameter omega
`eta`	a vector, the parameter eta
`priors`	an object of class 'mcmcPrior', see ?mcmcPrior

Value

returns the first and second derivatives of the prior

See Also

survspat, betapriorGauss, omegapriorGauss, etapriorGauss, indepGaussianprior, derivindepGaussianprior

distinfo

distinfo function

Description

Generic function for returning information about the class of baseline hazard functions employed.

Usage

```
distinfo(obj, ...)
```

Arguments

`obj`	an object
`...`	additional argument – currently there are none, but this is for extensibility

Value

method distinfo

See Also

distinfo.basehazardspec, exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

---

distinfo.basehazardspec *distinfo.basehazardspec function*

Description

A function to retrive information on the baseline hazard distribution of choice

Usage

```
## S3 method for class 'basehazardspec'
distinfo(obj, ...)
```

Arguments

`obj`	an object of class basehazardspec
`...`	additional arguments – currently there are none, but this is for extensibility

Value

a function returning information on the baseline hazard distribution of choice

See Also

exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

---

estimateY *estimateY function*

Description

A function to get an initial estimate of Y, to be used in calibrating the MCMC. Not for general use

Usage

```
estimateY(X, betahat, omegahat, surv, control)
```

Arguments

`X`	the design matrix containing covariate information
`betahat`	an estimate of beta
`omegahat`	an estimate of omega
`surv`	an object of class Surv
`control`	a list containg various control parameters for the MCMC and post-processing routines

Value

an estimate of Y, to be used in calibrating the MCMC

`etapriorGauss` *etapriorGauss function*

Description

A function to define Gaussian priors for eta. This function simply stores a vector of means and standard deviations to be passed to the main MCMC function, survspat.

Usage

```
etapriorGauss(mean, sd)
```

Arguments

`mean`	the prior mean, a vector of length 1 or more. 1 implies a common mean.
`sd`	the prior standard deviation, a vector of length 1 or more. 1 implies a common standard deviation.

Value

an object of class "etapriorGauss"

See Also

survspat, betapriorGauss, omegapriorGauss, etapriorGauss, indepGaussianprior, derivindepGaussianprior

Et_PP *Et_PP function*

Description

A function to compute an individual's approximate expected survival time using numerical integration. Note this appears to be unstable; the function is based on R's integrate function. Not intended for general use (yet!).

Usage

```
Et_PP(inputs)
```

Arguments

`inputs`	inputs for the function including the model matrix, frailties, fixed effects and the parameters of the baseline hazard derived from this model

Value

the expected survival time for the individual, obtained by numerical integration of the density function.

EvalCov *EvalCov function*

Description

This function is used to evaluate the covariance function within the MCMC run. Not intended for general use.

Usage

```
EvalCov(cov.model, u, parameters)
```

Arguments

`cov.model`	an object of class covmodel
`u`	vector of distances
`parameters`	vector of parameters

Value

method EvalCov

`ExponentialCovFct` *ExponentialCovFct function*

Description

A function to declare and also evaluate an exponential covariance function.

Usage

```
ExponentialCovFct()
```

Value

the exponential covariance function

See Also

SpikedExponentialCovFct, covmodel, CovarianceFct

`exponentialHaz` *exponentialHaz function*

Description

A function to define a parametric proportional hazards model where the baseline hazard is taken from the exponential model. This function returns an object inheriting class 'basehazardspec', list of functions 'distinfo', 'basehazard', 'gradbasehazard', 'hessbasehazard', 'cumbasehazard', 'gradcumbasehazard', 'hesscumbasehazard' and 'densityquantile'

Usage

```
exponentialHaz()
```

Details

The `distinfo` function is used to provide basic distribution specific information to other `spatsurv` functions. The user is required to provide the following information in the returned list: `npars`, the number of parameters in this distribution; `parnames`, the names of the parameters; `trans`, the transformation scale on which the priors will be provided; `itrans`, the inverse transformation function that will be applied to the parameters before the hazard, and other functions are evaluated; `jacobian`, the derivative of the inverse transformation function with respect to each of the parameters; and `hessian`, the second derivatives of the inverse transformation function with respect to each of the parameters – note that currently the package `spatsurv` only allows the use of functions where the parameters are transformed independently.

The `basehazard` function is used to evaluate the baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, `t` and returns a vector.

The `gradbasehazard` function is used to evaluate the gradient of the baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, `t`, and returns a matrix.

The `hessbasehazard` function is used to evaluate the Hessian of the baseline hazard function. It returns a function that accepts as input a vector of times, `t` and returns a list of hessian matrices corresponding to each `t`.

The `cumbasehazard` function is used to evaluate the cumulative baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, `t` and returns a vector.

The `gradcumbasehazard` function is used to evaluate the gradient of the cumulative baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, `t`, and returns a matrix.

The `hesscumbasehazard` function is used to evaluate the Hessian of the cumulative baseline hazard function. It returns a function that accepts as input a vector of times, `t` and returns a list of hessian matrices corresponding to each `t`.

The `densityquantile` function is used to return quantiles of the density function. This is NOT REQUIRED for running the MCMC, merely for us in post-processing with the `predict` function where `type` is 'densityquantile'. In the case of the Weibull model for the baseline hazard, it can be shown that the q-th quantile is:

Value

an object inheriting class 'basehazardspec'

See Also

tpowHaz, gompertzHaz, makehamHaz, weibullHaz

`FFTgrid` *FFTgrid function*

Description

A function to generate an FFT grid and associated quantities including cell dimensions, size of extended grid, centroids,

Usage

```
FFTgrid(spatialdata, cellwidth, ext)
```

Arguments

`spatialdata`	a SpatialPixelsDataFrame object
`cellwidth`	width of computational cells
`ext`	multiplying constant: the size of the extended grid: ext*M by ext*N

Value

a list

`fixedpars` *fixedpars function*

Description

A function to return the mcmc chains for the covariate effects

Usage

```
fixedpars(x)
```

Arguments

`x`	an object of class mcmcspatsurv

Value

the beta mcmc chains

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

---

`fixmatrix` *fixmatrix function*

Description

!! THIS FUNCTION IS NOT INTENDED FOR GENERAL USE !!

Usage

```
fixmatrix(mat)
```

Arguments

`mat`	a matrix

Details

A function to fix up an estimated covariance matrix using a VERY ad-hoc method.

Value

the fixed matrix

`frailtylag1` *frailtylag1 function*

Description

A function to produce a plot of, and return, the lag 1 (or higher, see argument 'lag') autocorrelation for each of the spatially correlated frailty chains

Usage

```
frailtylag1(object, plot = TRUE, lag = 1, ...)
```

Arguments

`object`	an object inheriting class mcmcspatsurv
`plot`	logical whether to plot the result, default is TRUE
`lag`	the lag to plot, the default is 1
`...`	other arguments to be passed to the plot function

Value

the lag 1 autocorrelation for each of the spatially correlated frailty chains

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

---

`GammafromY` *GammafromY function*

Description

A function to change Ys (spatially correlated noise) into Gammas (white noise). Used in the MALA algorithm.

Usage

```
GammafromY(Y, rootQeigs, mu)
```

Arguments

Y	Y matrix
rootQeigs	square root of the eigenvectors of the precision matrix
mu	parameter of the latent Gaussian field

Value

Gamma

GammaFromY_SPDE *GammaFromY_SPDE function*

Description

A function to go from Y to Gamma

Usage

```
GammaFromY_SPDE(Y, U, mu)
```

Arguments

Y	Y
U	upper Cholesky matrix
mu	the mean

Value

the value of Gamma for the given Y

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

2. Finn Lindgren, Havard Rue, Johan Lindstrom. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B 73(4)

gencens *gencens function*

Description

A function to generate observed times given a vector of true survival times and a vector of censoring times. Used in the simulation of survival data.

Usage

```
gencens(survtimes, censtimes, type = "right")
```

Arguments

`survtimes`	a vector of survival times
`censtimes`	a vector of censoring times for left or right censored data, 2-column matrix of censoring times for interval censoring (number of rows equal to the number of observations).
`type`	the type of censoring to generate can be 'right' (default), 'left' or 'interval'

Value

an object of class 'Surv', the censoring indicator is equal to 1 if the event is uncensored and 0 otherwise for right/left censored data, or for interval censored data, the indicator is 0 uncensored, 1 right censored, 2 left censored, or 3 interval censored.

getBackground *getBackground function*

Description

A function to

Usage

```
getBackground(poly, type = "stamen-toner")
```

Arguments

`poly`	X
`type`	X

Value

...

getBbasis *getBbasis function*

Description

A function returning the piecewise polynomial coefficients for a B-spline basis function i.e. the basis functions.

Usage

```
getBbasis(x, knots, degree)
```

Arguments

`x`	a vector of data
`knots`	a vector of knots in ascending order. The first and last knots must be respectively the minimum and maximum of x.
`degree`	the degree of the spline

Value

the knots and the piecewise polynomial coefficients for a B-spline basis function i.e. the basis functions.

getcov *getcov function*

Description

A function to return the covariance from a model based on the randomFields covariance functions. Not intended for general use.

Usage

```
getcov(u, sigma, phi, model, pars)
```

Arguments

`u`	distance
`sigma`	variance parameter
`phi`	scale parameter
`model`	correlation type, see ?CovarianceFct
`pars`	vector of additional parameters for certain classes of covariance function (eg Matern), these must be supplied in the order given in ?CovarianceFct and are not estimated

Value

this is just a wrapper for CovarianceFct

---

getgrd *getgrd function*

Description

A function to create a regular grid over an observation window in order to model the spatial randome effects as a Gaussian Markov random field.

Usage

```
getgrd(shape, cellwidth)
```

Arguments

`shape`	an object of class SpatialPolygons or SpatialPolygonsDataFrame
`cellwidth`	a scalar, the width of the grid cells

Value

a SpatialPolygons object: the grid on which prediction of the spatial effects will occur

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665
2. Finn Lindgren, Havard Rue, Johan Lindstrom. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B 73(4)

---

getGrid *getGrid function*

Description

A function to extract and return the computational grid from a gridded analysis.

Usage

```
getGrid(mod, returnclass = "SpatialPolygonsDataFrame")
```

Arguments

`mod`	an object of class mcmcspatsurv, returned by the function survspat
`returnclass`	the class of object to return, default is a'SpatialPolygonsDataFrame'. Other options are 'raster', which returns a raster brick; or 'SpatialPixelsDataFrame'

Value

a SpatialPolygonsDataFrame in which Monte Carlo expectations can be stored and later plotted.

`getleneta` *getleneta function*

Description

A function to compute the length of eta

Usage

```
getleneta(cov.model)
```

Arguments

`cov.model`	a covariance model

Value

the length of eta

`getparranges` *getparranges function*

Description

A function to extract parameter ranges for creating a grid on which to evaluate the log-posterior, used in calibrating the MCMC. This function is not intended for general use.

Usage

```
getparranges(priors, leneta, mult = 1.96)
```

Arguments

`priors`	an object of class mcmcPriors
`leneta`	the length of eta passed to the function
`mult`	defaults to 1.96 so the grid formed will be mean plus/minus 1.96 times the standard deviation

Value

an appropriate range used to calibrate the MCMC: the mean of the prior for eta plus/minus 1.96 times the standard deviation

---

`getsurvdata` *getsurvdata function*

Description

A function to return the survival data from an object of class mcmcspatsurv. This function is not intended for general use.

Usage

```
getsurvdata(x)
```

Arguments

`x`	an object of class mcmcspatsurv

Value

the survival data from an object of class mcmcspatsurv

`gompertzHaz` *gompertzHaz function*

Description

A function to define a parametric proportional hazards model where the baseline hazard is taken from a Gompertz model. This function returns an object inheriting class 'basehazardspec', list of functions 'distinfo', 'basehazard', 'gradbasehazard', 'hessbasehazard', 'cumbasehazard', 'gradcumbasehazard', 'hesscumbasehazard' and 'densityquantile'

Usage

```
gompertzHaz()
```

Details

The distinfo function is used to provide basic distribution specific information to other spatsurv functions. The user is required to provide the following information in the returned list: npars, the number of parameters in this distribution; parnames, the names of the parameters; trans, the transformation scale on which the priors will be provided; itrans, the inverse transformation function that will be applied to the parameters before the hazard, and other functions are evaluated; jacobian, the derivative of the inverse transformation function with respect to each of the parameters; and hessian, the second derivatives of the inverse transformation function with respect to each of the parameters – note that currently the package spatsurv only allows the use of functions where the parameters are transformed independently.

The basehazard function is used to evaluate the baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, t and returns a vector.

The gradbasehazard function is used to evaluate the gradient of the baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, t, and returns a matrix.

The hessbasehazard function is used to evaluate the Hessian of the baseline hazard function. It returns a function that accepts as input a vector of times, t and returns a list of hessian matrices corresponding to each t.

The cumbasehazard function is used to evaluate the cumulative baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, t and returns a vector.

The gradcumbasehazard function is used to evaluate the gradient of the cumulative baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, t, and returns a matrix.

The hesscumbasehazard function is used to evaluate the Hessian of the cumulative baseline hazard function. It returns a function that accepts as input a vector of times, t and returns a list of hessian matrices corresponding to each t.

The densityquantile function is used to return quantiles of the density function. This is NOT REQUIRED for running the MCMC, merely for us in post-processing with the predict function where type is 'densityquantile'. In the case of the Weibull model for the baseline hazard, it can be shown that the q-th quantile is:

Value

an object inheriting class 'basehazardspec'

See Also

tpowHaz, exponentialHaz, makehamHaz, weibullHaz

gradbasehazard *gradbasehazard function*

Description

Generic function for computing the gradient of the baseline hazard

Usage

```
gradbasehazard(obj, ...)
```

Arguments

obj	an object
...	additional arguments – currently there are none, but this is for extensibility

Value

method gradbasehazard

See Also

gradbasehazard.basehazardspec, exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

gradbasehazard.basehazardspec *gradbasehazard.basehazardspec function*

Description

A function to retrieve the gradient of the baseline hazard function

Usage

```
## S3 method for class 'basehazardspec'
gradbasehazard(obj, ...)
```

Arguments

obj	an object of class basehazardspec
...	additional arguments – currently there are none, but this is for extensibility

Value

a function returning the gradient of the baseline hazard

See Also

exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

`gradcumbasehazard` *gradcumbasehazard function*

Description

Generic function for computing the gradient of the cumulative baseline hazard

Usage

```
gradcumbasehazard(obj, ...)
```

Arguments

`obj`	an object
`...`	additional arguments – currently there are none, but this is for extensibility

Value

method gradcumbasehazard

See Also

gradcumbasehazard.basehazardspec, exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

---

`gradcumbasehazard.basehazardspec` *gradcumbasehazard.basehazardspec function*

Description

A function to retrieve the gradient of the cumulative baseline hazard function

Usage

```
## S3 method for class 'basehazardspec'
gradcumbasehazard(obj, ...)
```

Arguments

`obj`	an object of class basehazardspec
`...`	additional arguments – currently there are none, but this is for extensibility

Value

a function returning the gradient of the cumulative baseline hazard

See Also

exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

`grid2spdf` *grid2spdf function*

Description

A function to convert a regular (x,y) grid of centroids into a SpatialPoints object

Usage

```
grid2spdf(xgrid, ygrid, proj4string = CRS(as.character(NA)))
```

Arguments

`xgrid`	vector of x centroids (equally spaced)
`ygrid`	vector of x centroids (equally spaced)
`proj4string`	an optional proj4string, projection string for the grid, set using the function CRS

Value

a SpatialPolygonsDataFrame

`grid2spix` *grid2spix function*

Description

A function to convert a regular (x,y) grid of centroids into a SpatialPixels object

Usage

```
grid2spix(xgrid, ygrid, proj4string = CRS(as.character(NA)))
```

Arguments

`xgrid`	vector of x centroids (equally spaced)
`ygrid`	vector of x centroids (equally spaced)
`proj4string`	an optional proj4string, projection string for the grid, set using the function CRS

Value

a SpatialPixels object

`grid2spts` *grid2spts function*

Description

A function to convert a regular (x,y) grid of centroids into a SpatialPoints object

Usage

```
grid2spts(xgrid, ygrid, proj4string = CRS(as.character(NA)))
```

Arguments

`xgrid`	vector of x centroids (equally spaced)
`ygrid`	vector of x centroids (equally spaced)
`proj4string`	an optional proj4string, projection string for the grid, set using the function CRS

Value

a SpatialPoints object

`gridY` *gridY function*

Description

A function to put estimated individual Y's onto a grid

Usage

```
gridY(Y, control)
```

Arguments

`Y`	estimate of Y
`control`	control parameters

Value

...

gridY_polygonal *gridY_polygonal function*

Description

A function to put estimated individual Y's onto a grid

Usage

```
gridY_polygonal(Y, control)
```

Arguments

`Y`	estimate of Y
`control`	control parameters

Value

...

guess_t *guess_t function*

Description

A function to get an initial guess of the failure time t, to be used in calibrating the MCMC. Not for general use

Usage

```
guess_t(surv)
```

Arguments

`surv`	an object of class Surv

Value

a guess at the failure times

hasNext — *generic hasNext method*

Description

test if an iterator has any more values to go

Usage

```
hasNext(obj)
```

Arguments

obj	an iterator

hasNext.iter — *hasNext.iter function*

Description

method for iter objects test if an iterator has any more values to go

Usage

```
## S3 method for class 'iter'
hasNext(obj)
```

Arguments

obj	an iterator

hazardexceedance — *hazardexceedance function*

Description

A function to compute exceedance probabilities for the spatially correlated frailties.

Usage

```
hazardexceedance(threshold, direction = "upper")
```

Arguments

threshold	vector of thresholds
direction	default is "upper" which will calculate P(Y>threshold), alternative is "lower", which will calculate P(Y<threshold)

Value

a function that can be passed to the function MCE in order to compute the exceedance probabilities

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE,

---

hazardpars *hazardpars function*

Description

A function to return the mcmc chains for the hazard function parameters

Usage

```
hazardpars(x)
```

Arguments

x	an object of class mcmcspatsurv

Value

the omega mcmc chains

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance hazard_PP *hazard_PP function*

Description

A function to compute an individual's hazard function.

Usage

```
hazard_PP(inputs)
```

Arguments

inputs	inputs for the function including the model matrix, frailties, fixed effects and the parameters of the baseline hazard derived from this model

Value

the hazard function for the individual

hessbasehazard *hessbasehazard function*

Description

Generic function for computing the hessian of the baseline hazard

Usage

```
hessbasehazard(obj, ...)
```

Arguments

obj	an object
...	additional arguments – currently there are none, but this is for extensibility

Value

method hessbasehazard

See Also

hessbasehazard.basehazardspec, exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

hessbasehazard.basehazardspec

hessbasehazard.basehazardspec function

Description

A function to retrieve the Hessian of the baseline hazard function

Usage

```
## S3 method for class 'basehazardspec'
hessbasehazard(obj, ...)
```

Arguments

obj	an object of class basehazardspec
...	additional arguments – currently there are none, but this is for extensibility

Value

a function returning the Hessian of the baseline hazard

See Also

exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

hesscumbasehazard

hesscumbasehazard function

Description

Generic function for computing the Hessian of the cumulative baseline hazard

Usage

```
hesscumbasehazard(obj, ...)
```

Arguments

obj	an object
...	additional arguments – currently there are none, but this is for extensibility

Value

method hesscumbasehazard

See Also

hesscumbasehazard.basehazardspec, exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

---

`hesscumbasehazard.basehazardspec` *hesscumbasehazard.basehazardspec function*

Description

A function to retrieve the hessian of the cumulative baseline hazard function

Usage

```
## S3 method for class 'basehazardspec'
hesscumbasehazard(obj, ...)
```

Arguments

`obj`	an object of class basehazardspec
`...`	additional arguments – currently there are none, but this is for extensibility

Value

a function returning the hessian of the cumulative baseline hazard

See Also

exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz

`indepGaussianprior` *indepGaussianprior function*

Description

A function for evaluating the log of an independent Gaussian prior for a given set of parameter values.

Usage

```
indepGaussianprior(beta = NULL, omega = NULL, eta = NULL, priors)
```

Arguments

`beta`	parameter beta at which prior is to be evaluated
`omega`	parameter omega at which prior is to be evaluated
`eta`	parameter eta at which prior is to be evaluated
`priors`	an object of class mcmcPriors, see ?mcmcPriors

Value

the log of the prior evaluated at the given parameter values

See Also

survspat, betapriorGauss, omegapriorGauss, etapriorGauss, indepGaussianprior, derivindepGaussianprior

---

`inference.control` *inference.control function*

Description

A function to control inferential settings. This function is used to set parameters for more advanced use of spatsurv.

Usage

```
inference.control(gridded = FALSE, cellwidth = NULL, ext = 2,
  optimcontrol = NULL, hessian = FALSE, plotcal = FALSE,
  timeonlyMCMC = FALSE)
```

Arguments

`gridded`	logical. Whether to perform compuation on a grid. Default is FALSE.
`cellwidth`	the width of computational cells to use
`ext`	integer the number of times to extend the computational grid by in order to perform compuitation. The default is 2.
`optimcontrol`	a list of optional arguments to be passed to optim for non-spatial models
`hessian`	whether to return a numerical hessian. Set this to TRUE for non-spatial models. equal to the number of parameters of the baseline hazard
`plotcal`	logical, whether to produce plots of the MCMC calibration process, this is a technical option and should onyl be set to TRUE if poor mixing is evident (the printed h is low), then it is also useful to use a graphics device with multiple plotting windows.
`timeonlyMCMC`	logical, whether to only time the MCMC part of the algorithm, or whether to include in the reported running time the time taken to calibrate the method (default)

Value

returns parameters to be used in the function survspat

See Also

survspat

`invtransformweibull` *invtransformweibull function*

Description

A function to transform estimates of the (alpha, lambda) parameters of the weibull baseline hazard function, so they are commensurate with R's inbuilt density functions, (shape, scale).

Usage

```
invtransformweibull(x)
```

Arguments

`x`	a vector of paramters

Value

the transformed parameters. For the weibull model, this transforms 'shape' 'scale' (see ?dweibull) to 'alpha' and 'lambda' for the MCMC

`is.burnin` *is this a burn-in iteration?*

Description

if this mcmc iteration is in the burn-in period, return TRUE

Usage

```
is.burnin(obj)
```

Arguments

`obj`	an mcmc iterator

Value

TRUE or FALSE

`is.retain` *do we retain this iteration?*

Description

if this mcmc iteration is one not thinned out, this is true

Usage

```
is.retain(obj)
```

Arguments

`obj`	an mcmc iterator

Value

TRUE or FALSE

`iteration` *iteration number*

Description

within a loop, this is the iteration number we are currently doing.

Usage

```
iteration(obj)
```

Arguments

`obj`	an mcmc iterator

Details

get the iteration number

Value

integer iteration number, starting from 1.

`logPosterior` *logPosterior function*

Description

A function to evaluate the log-posterior of a spatial parametric proportional hazards model. Not intended for general use.

Usage

```
logPosterior(surv, X, beta, omega, eta, gamma, priors, cov.model, u, control,
  gradient = FALSE, hessian = FALSE)
```

Arguments

`surv`	an object of class Surv
`X`	the design matrix, containing covariate information
`beta`	parameter beta
`omega`	parameter omega
`eta`	parameter eta
`gamma`	parameter gamma
`priors`	the priors, an object of class 'mcmcPriors'
`cov.model`	the spatial covariance model
`u`	vector of interpoint distances
`control`	a list containg various control parameters for the MCMC and post-processing routines
`gradient`	logical whether to evaluate the gradient
`hessian`	logical whether to evaluate the Hessian

Value

evaluates the log-posterior and the gradient and hessian, if required.

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

logPosterior_gridded *logPosterior_gridded function*

Description

A function to evaluate the log-posterior of a spatial parametric proportional hazards model using gridded Y. Not intended for general use.

Usage

```
logPosterior_gridded(surv, X, beta, omega, eta, gamma, priors, cov.model, u,
  control, gradient = FALSE, hessian = FALSE)
```

Arguments

surv	an object of class Surv
X	the design matrix, containing covariate information
beta	parameter beta
omega	parameter omega
eta	parameter eta
gamma	parameter gamma
priors	the priors, an object of class 'mcmcPriors'
cov.model	the spatial covariance model
u	vector of interpoint distances
control	a list containg various control parameters for the MCMC and post-processing routines
gradient	logical whether to evaluate the gradient
hessian	logical whether to evaluate the Hessian

Value

evaluates the log-posterior and the gradient and hessian, if required.

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

`logPosterior_polygonal` *logPosterior_polygonal function*

Description

A function to evaluate the log-posterior of a spatial parametric proportional hazards model. Not intended for general use.

Usage

```
logPosterior_polygonal(surv, X, beta, omega, eta, gamma, priors, cov.model, u,
  control, gradient = FALSE, hessian = FALSE)
```

Arguments

`surv`	an object of class Surv
`X`	the design matrix, containing covariate information
`beta`	parameter beta
`omega`	parameter omega
`eta`	parameter eta
`gamma`	parameter gamma
`priors`	the priors, an object of class 'mcmcPriors'
`cov.model`	the spatial covariance model
`u`	vector of interpoint distances
`control`	a list containg various control parameters for the MCMC and post-processing routines
`gradient`	logical whether to evaluate the gradient
`hessian`	logical whether to evaluate the Hessian

Value

evaluates the log-posterior and the gradient and hessian, if required.

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

`logPosterior_SPDE` *logPosterior_SPDE function*

Description

A function to evaluate the log-posterior of a spatial parametric proportional hazards model. Not intended for general use.

Usage

```
logPosterior_SPDE(surv, X, beta, omega, eta, gamma, priors, cov.model, u,
  control, gradient = FALSE, hessian = FALSE)
```

Arguments

`surv`	an object of class Surv
`X`	the design matrix, containing covariate information
`beta`	parameter beta
`omega`	parameter omega
`eta`	parameter eta
`gamma`	parameter gamma
`priors`	the priors, an object of class 'mcmcPriors'
`cov.model`	the spatial covariance model
`u`	vector of interpoint distances
`control`	a list containg various control parameters for the MCMC and post-processing routines
`gradient`	logical whether to evaluate the gradient
`hessian`	logical whether to evaluate the Hessian

Value

evaluates the log-posterior and the gradient and hessian, if required.

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

2. Finn Lindgren, Havard Rue, Johan Lindstrom. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B 73(4)

`loop.mcmc` *loop over an iterator*

Description

useful for testing progress bars

Usage

```
loop.mcmc(object, sleep = 1)
```

Arguments

`object`	an mcmc iterator
`sleep`	pause between iterations in seconds

`makehamHaz` *makehamHaz function*

Description

A function to define a parametric proportional hazards model where the baseline hazard is taken from the Gompertz-Makeham model. This function returns an object inheriting class 'basehazardspec', list of functions 'distinfo', 'basehazard', 'gradbasehazard', 'hessbasehazard', 'cumbasehazard', 'gradcumbasehazard', 'hesscumbasehazard' and 'densityquantile'

Usage

```
makehamHaz()
```

Details

The `distinfo` function is used to provide basic distribution specific information to other `spatsurv` functions. The user is required to provide the following information in the returned list: `npars`, the number of parameters in this distribution; `parnames`, the names of the parameters; `trans`, the transformation scale on which the priors will be provided; `itrans`, the inverse transformation function that will be applied to the parameters before the hazard, and other functions are evaluated; `jacobian`, the derivative of the inverse transformation function with respect to each of the parameters; and `hessian`, the second derivatives of the inverse transformation function with respect to each of the parameters – note that currently the package `spatsurv` only allows the use of functions where the parameters are transformed independently.

The `basehazard` function is used to evaluate the baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, `t` and returns a vector.

The gradbasehazard function is used to evaluate the gradient of the baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, t, and returns a matrix.

The hessbasehazard function is used to evaluate the Hessian of the baseline hazard function. It returns a function that accepts as input a vector of times, t and returns a list of hessian matrices corresponding to each t.

The cumbasehazard function is used to evaluate the cumulative baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, t and returns a vector.

The gradcumbasehazard function is used to evaluate the gradient of the cumulative baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, t, and returns a matrix.

The hesscumbasehazard function is used to evaluate the Hessian of the cumulative baseline hazard function. It returns a function that accepts as input a vector of times, t and returns a list of hessian matrices corresponding to each t.

The densityquantile function is used to return quantiles of the density function. This is NOT REQUIRED for running the MCMC, merely for us in post-processing with the predict function where type is 'densityquantile'. In the case of the Weibull model for the baseline hazard, it can be shown that the q-th quantile is:

Value

an object inheriting class 'basehazardspec'

See Also

tpowHaz, exponentialHaz, gompertzHaz, weibullHaz

---

maxlikparamPHsurv *maxlikparamPHsurv function*

Description

A function to get initial estimates of model parameters using maximum likelihood. Not intended for general purose use.

Usage

```
maxlikparamPHsurv(surv, X, control)
```

Arguments

surv	an object of class Surv
X	the design matrix, containing covariate information
control	a list containg various control parameters for the MCMC and post-processing routines

Value

initial estimates of the parameters

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

`MCE` *MCE function*

Description

A function to compute Monte Carlo expectations from an object inheriting class mcmcspatsurv

Usage

```
MCE(object, fun)
```

Arguments

`object`	an object inheriting class mcmcspatsurv
`fun`	a function with arguments beta, omega, eta and Y

Value

the Monte Carlo mean of the function over the posterior.

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, hazardexceedance

mcmcLoop *iterator for MCMC loops*

Description

control an MCMC loop with this iterator

Usage

```
mcmcLoop(N, burnin, thin, trim = TRUE, progressor = mcmcProgressPrint)
```

Arguments

N	number of iterations
burnin	length of burn-in
thin	frequency of thinning
trim	whether to cut off iterations after the last retained iteration
progressor	a function that returns a progress object

mcmcpars *mcmcpars function*

Description

A function for setting MCMC options.

Usage

```
mcmcpars(nits, burn, thin, inits = NULL, adaptivescheme = NULL)
```

Arguments

nits	numer of iterations,
burn	length of burnin
thin	thinning parameter eg operated on chain every 'thin' iteration (eg store output or compute some posterior functional)
inits	NOT CURRENTLY IN USE
adaptivescheme	NOT CURRENTLY IN USE

Value

mcmc parameters

mcmcPriors *mcmcPriors function*

Description

A function to define priors for the MCMC.

Usage

```
mcmcPriors(betaprior = NULL, omegaprior = NULL, etaprior = NULL,
  call = NULL, derivative = NULL)
```

Arguments

betaprior	prior for beta, the covariate effects
omegaprior	prior for omega, the parameters of the baseline hazard
etaprior	prior for eta, the parameters of the latent field
call	function to evaluate the log-prior e.g. logindepGaussianprior
derivative	function to evaluate the first and second derivatives of the prior

Details

The package `spatsurv` only provides functionality for the built-in Gaussian priors. However, the choice of prior is extensible by the user by creating functions similar to the functions `betapriorGauss`, `omegapriorGauss`, `etapriorGauss`, `indepGaussianprior` and `derivindepGaussianprior`: the first three of which provide a mechanism for storing and retrieving the parameters of the priors; the fourth, a function for evaluating the log of the prior for a given set of parameter values; and the fifth, a function for evaluating the first and second derivatives of the log of the prior. It is assumed that parameters are a priori independent. The user interested in using other priors is encouraged to look at the structure of the five functions mentioned above.

Value

an object of class mcmcPriors

See Also

survspat, betapriorGauss, omegapriorGauss, etapriorGauss, indepGaussianprior, derivindepGaussianprior

`mcmcProgressNone` *null progress monitor*

Description

a progress monitor that does nothing

Usage

```
mcmcProgressNone(mcmcloop)
```

Arguments

`mcmcloop`	an mcmc loop iterator

Value

a progress monitor

`mcmcProgressPrint` *printing progress monitor*

Description

a progress monitor that prints each iteration

Usage

```
mcmcProgressPrint(mcmcloop)
```

Arguments

`mcmcloop`	an mcmc loop iterator

Value

a progress monitor

mcmcProgressTextBar   *text bar progress monitor*

Description

a progress monitor that uses a text progress bar

Usage

```
mcmcProgressTextBar(mcmcloop)
```

Arguments

mcmcloop	an mcmc loop iterator

Value

a progress monitor

midpts   *midpts function*

Description

A function to compute the midpoints of a vector

Usage

```
midpts(x)
```

Arguments

x	a vector

Value

the midpoints, a vector of length length(x)-1 `neighLocs` *neighLocs function*

Description

A function used in the computation of neighbours on non-rectangular grids. Not intended for general use.

Usage

```
neighLocs(coord, cellwidth, order)
```

Arguments

`coord`	coordinate of interest
`cellwidth`	a scalar, the width of the grid cells
`order`	the order of the SPDE approximation: see Lindgren et al 2011 for details

Value

coordinates of centroids of neighbours

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665
2. Finn Lindgren, Havard Rue, Johan Lindstrom. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B 73(4)

`neighOrder` *neighOrder function*

Description

A function to compute the order of a set of neighbours. Not intended for general use.

Usage

```
neighOrder(neighlocs)
```

Arguments

`neighlocs`	an object created by the function neighLocs

Value

the neighbour orders

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665
2. Finn Lindgren, Havard Rue, Johan Lindstrom. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B 73(4)

---

nextStep — *next step of an MCMC chain*

Description

just a wrapper for nextElem really.

Usage

```
nextStep(object)
```

Arguments

`object`	an mcmc loop object

NonSpatialLogLikelihood_or_gradient — *NonSpatialLogLikelihood_or_gradient function*

Description

A function to evaluate the log-likelihood of a non-spatial parametric proportional hazards model. Not intended for general use.

Usage

```
NonSpatialLogLikelihood_or_gradient(surv, X, beta, omega, control,
  loglikelihood, gradient)
```

Arguments

surv	an object of class Surv
X	the design matrix, containing covariate information
beta	parameter beta
omega	parameter omega
control	a list containg various control parameters for the MCMC and post-processing routines
loglikelihood	logical whether to evaluate the log-likelihood
gradient	logical whether to evaluate the gradient

Value

...

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

---

omegapriorGauss *omegapriorGauss function*

Description

A function to define Gaussian priors for omega. This function simply stores a vector of means and standard deviations to be passed to the main MCMC function, survspat.

Usage

```
omegapriorGauss(mean, sd)
```

Arguments

mean	the prior mean, a vector of length 1 or more. 1 implies a common mean.
sd	the prior standard deviation, a vector of length 1 or more. 1 implies a common standard deviation.

Value

an object of class "omegapriorGauss"

See Also

survspat, betapriorGauss, omegapriorGauss, etapriorGauss, indepGaussianprior, derivindepGaussianprior

plotsurv *plotsurv function*

Description

A function to produce a 2-D plot of right censored spatial survival data.

Usage

```
plotsurv(spp, ss, maxcex = 1, transform = identity, background = NULL,
  eventpt = 19, eventcol = "red", censpt = "+", censcol = "black",
  xlim = NULL, ylim = NULL, xlab = NULL, ylab = NULL, add = FALSE,
  ...)
```

Arguments

`spp`	A spatial points data frame
`ss`	A Surv object (with right-censoring)
`maxcex`	maximum size of dots default is equavalent to setting cex equal to 1
`transform`	optional transformation to apply to the data, a function, for example 'sqrt'
`background`	a background object to plot default is null, which gives a blamk background note that if non-null, the parameters xlim and ylim will be derived from this object.
`eventpt`	The type of point to illustrate events, default is 19 (see ?pch)
`eventcol`	the colour of events, default is black
`censpt`	The type of point to illustrate events, default is "+" (see ?pch)
`censcol`	the colour of censored observations, default is red
`xlim`	optional x-limits of plot, default is to choose this automatically
`ylim`	optional y-limits of plot, default is to choose this automatically
`xlab`	label for x-axis
`ylab`	label for y-axis
`add`	logical, whether to add the survival plot on top of an existing plot, default is FALSE, which produces a plot in a new device
`...`	other arguments to pass to plot

Value

Plots the survival data non-censored observations appear as dots and censored observations as crosses. The size of the dot is proportional to the observed time.

polyadd *polyadd function*

Description

A function to add two polynomials in the form of vectors of coefficients. The first element of the vector being the constant (order 0) term

Usage

```
polyadd(poly1, poly2)
```

Arguments

poly1	a vector of coefficients for the first polynomial of length degree plus 1
poly2	a vector of coefficients for the second polynomial of length degree plus 1

Value

the coefficients of the sum of poly1 and poly2

polymult *polymult function*

Description

A function to multiply two polynomials in the form of vectors of coefficients. The first element of the vector being the constant (order 0) term

Usage

```
polymult(poly1, poly2)
```

Arguments

poly1	a vector of coefficients for the first polynomial of length degree plus 1
poly2	a vector of coefficients for the second polynomial of length degree plus 1

Value

the coefficients of the product of poly1 and poly2 posteriorcov *posteriorcov function*

Description

A function to produce a plot of the posterior covariance function with upper and lower quantiles.

Usage

```
posteriorcov(x, probs = c(0.025, 0.5, 0.975), rmax = NULL, n = 100,
  plot = TRUE, bw = FALSE, ...)
```

Arguments

x	an object of class mcmcspatsurv
probs	vector of probabilities to be fed to quantile function
rmax	maximum distance in space to compute this distance up to
n	the number of points at which to evaluate the posterior covariance.
plot	whether to plot the result
bw	Logical. Plot in black/white/greyscale? Default is to produce a colour plot. Useful for producing plots for journals that do not accept colour plots.
...	other arguments to be passed to matplot function

Value

produces a plot of the posterior spatial covariance function.

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, MCE, hazardexceedance

predict.mcmcspatsurv *predict.mcmcspatsurv function*

Description

A function to produce predictions from MCMC output. These could include quantiles of the individual density, survival or hazard functions or quantiles of the density function (if available analytically).

Usage

```
## S3 method for class 'mcmcspatsurv'
predict(object, type = "density", t = NULL,
  n = 110, indx = NULL, probs = c(0.025, 0.5, 0.975), plot = TRUE,
  pause = TRUE, bw = FALSE, ...)
```

Arguments

`object`	an object of class mcmcspatsurv
`type`	can be "density", "hazard", "survival" or "densityquantile". Default is "density". Note that "densityquantile" is not always analytically tractable for some choices of baseline hazard function.
`t`	optional vector of times at which to compute the quantiles, Defult is NULL, in which case a uniformly spaced vector of length n from 0 to the maximum time is used
`n`	the number of points at which to compute the quantiles if t is NULL
`indx`	the index number of a particular individual or vector of indices of individuals for which the quantiles should be produced
`probs`	vector of probabilities
`plot`	whether to plot the result
`pause`	logical whether to pause between plots, the default is TRUE
`bw`	Logical. Plot in black/white/greyscale? Default is to produce a colour plot. Useful for producing plots for journals that do not accept colour plots.
`...`	other arguments, not used here

Value

the required predictions

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, priorposterior, posteriorcov, MCE, hazardexceedance

---

`print.mcmc` *print.mcmc function*

Description

print method print an mcmc iterator's details

Usage

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

Arguments

x	a mcmc iterator
...	other args

---

print.mcmcspatsurv *print.mcmcspatsurv function*

Description

A function to print summary tables from an MCMC run

Usage

```
## S3 method for class 'mcmcspatsurv'
print(x, probs = c(0.5, 0.025, 0.975), digits = 3,
  scientific = -3, ...)
```

Arguments

x	an object inheriting class mcmcspatsurv
probs	vector of quantiles to return
digits	see help file ?format
scientific	see help file ?format
...	additional arguments, not used here

Value

prints summary tables to the console

See Also

quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

print.mlspatsurv *print.mlspatsurv function*

Description

A function to print summary tables from an MCMC run

Usage

```
## S3 method for class 'mlspatsurv'
print(x, probs = c(0.5, 0.025, 0.975), digits = 3,
  scientific = -3, ...)
```

Arguments

x	an object inheriting class mcmcspatsurv
probs	vector of quantiles to return
digits	see help file ?format
scientific	see help file ?format
...	additional arguments, not used here

Value

prints summary tables to the console

See Also

quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

priorposterior *priorposterior function*

Description

A function to produce plots of the prior (which shows as a red line) and posterior (showing as a histogram)

Usage

```
priorposterior(x, breaks = 30, ylab = "Density", main = "",
  pause = TRUE, bw = FALSE, ...)
```

Arguments

`x`	an object inheriting class mcmcspatsurv
`breaks`	see ?hist
`ylab`	optional y label
`main`	optional title
`pause`	logical whether to pause between plots, the default is TRUE
`bw`	Logical. Plot in black/white/greyscale? Default is to produce a colour plot. Useful for producing plots for journals that do not accept colour plots.
`...`	other arguments passed to the hist function

Value

plots of the prior (red line) and posterior (histogram).

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, posteriorcov, MCE, hazardexceedance

---

`proposalVariance` *proposalVariance function*

Description

A function to compute an approximate scaling matrix for the MCMC algorithm. Not intended for general use.

Usage

```
proposalVariance(X, surv, betahat, omegahat, Yhat, priors, cov.model, u,
  control)
```

Arguments

`X`	the design matrix, containing covariate information
`surv`	an object of class Surv
`betahat`	an estimate of beta
`omegahat`	an estimate of omega
`Yhat`	an estimate of Y
`priors`	the priors
`cov.model`	the spatial covariance model
`u`	a vector of pairwise distances
`control`	a list containg various control parameters for the MCMC and post-processing routines

Value

an estimate of eta and also an approximate scaling matrix for the MCMC

`proposalVariance_gridded` *proposalVariance_gridded function*

Description

A function to compute an approximate scaling matrix for the MCMC algorithm. Not intended for general use.

Usage

```
proposalVariance_gridded(X, surv, betahat, omegahat, Yhat, priors, cov.model, u,
  control)
```

Arguments

`X`	the design matrix, containing covariate information
`surv`	an object of class Surv
`betahat`	an estimate of beta
`omegahat`	an estimate of omega
`Yhat`	an estimate of Y
`priors`	the priors
`cov.model`	the spatial covariance model
`u`	a vector of pairwise distances
`control`	a list containg various control parameters for the MCMC and post-processing routines

Value

an estimate of eta and also an approximate scaling matrix for the MCMC

proposalVariance_polygonal *proposalVariance_polygonal function*

Description

A function to compute an approximate scaling matrix for the MCMC algorithm. Not intended for general use.

Usage

```
proposalVariance_polygonal(X, surv, betahat, omegahat, Yhat, priors, cov.model,
  u, control)
```

Arguments

X	the design matrix, containing covariate information
surv	an object of class Surv
betahat	an estimate of beta
omegahat	an estimate of omega
Yhat	an estimate of Y
priors	the priors
cov.model	the spatial covariance model
u	a vector of pairwise distances
control	a list containg various control parameters for the MCMC and post-processing routines

Value

an estimate of eta and also an approximate scaling matrix for the MCMC

proposalVariance_SPDE *proposalVariance_SPDE function*

Description

A function to compute an approximate scaling matrix for the MCMC algorithm. Not intended for general use.

Usage

```
proposalVariance_SPDE(X, surv, betahat, omegahat, Yhat, priors, cov.model, u,
  control)
```

Arguments

X	the design matrix, containing covariate information
surv	an object of class Surv
betahat	an estimate of beta
omegahat	an estimate of omega
Yhat	an estimate of Y
priors	the priors
cov.model	the spatial covariance model
u	a vector of pairwise distances
control	a list containg various control parameters for the MCMC and post-processing routines

Value

an estimate of eta and also an approximate scaling matrix for the MCMC

QuadApprox *QuadApprox function*

Description

A function to compute the second derivative of a function (of several real variables) using a quadratic approximation on a grid of points defined by the list argRanges. Also returns the local maximum.

Usage

```
QuadApprox(fun, npts, argRanges, plot = FALSE, ...)
```

Arguments

fun	a function
npts	integer number of points in each direction
argRanges	a list of ranges on which to construct the grid for each parameter
plot	whether to plot the quadratic approximation of the posterior (for two-dimensional parameters only)
...	other arguments to be passed to fun

Value

a 2 by 2 matrix containing the curvature at the maximum and the (x,y) value at which the maximum occurs

quantile.mcmcspatsurv *quantile.mcmcspatsurv function*

Description

A function to extract quantiles of the parameters from an mcmc run

Usage

```
## S3 method for class 'mcmcspatsurv'
quantile(x, probs = c(0.025, 0.5, 0.975), ...)
```

Arguments

`x`	an object inheriting class mcmcspatsurv
`probs`	vector of probabilities
`...`	other arguments to be passed to the function, not used here

Value

quantiles of model parameters

See Also

print.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

quantile.mlspatsurv *quantile.mlspatsurv function*

Description

A function to extract quantiles of the parameters from an mcmc run

Usage

```
## S3 method for class 'mlspatsurv'
quantile(x, probs = c(0.025, 0.5, 0.975), ...)
```

Arguments

`x`	an object inheriting class mcmcspatsurv
`probs`	vector of probabilities
`...`	other arguments to be passed to the function, not used here

Value

quantiles of model parameters

See Also

print.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

---

randompars — *randompars function*

Description

A function to return the mcmc chains for the spatially correlated frailties

Usage

```
randompars(x)
```

Arguments

x	an object of class mcmcspatsurv

Value

the Y mcmc chains

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

---

reconstruct.bs — *reconstruct.bs function*

Description

When bs(varname) has been used in the formula of a model, this function can be used to reconstruct the posterior relative risk of that parameter over time.

Usage

```
reconstruct.bs(mod, varname, probs = c(0.025, 0.975), bw = FALSE,
  xlab = NULL, ylab = NULL, plot = TRUE, ...)
```

Arguments

mod	model output, created by function survspat
varname	name of the variable modelled by a B-spline
probs	upper and lower quantiles for confidence regions to plot> The default is c(0.025,0.975).
bw	Logical. Plot in black/white/greyscale? Default is to produce a colour plot. Useful for producing plots for journals that do not accept colour plots.
xlab	label for x axis, there is a sensible default
ylab	label for y axis, there is a sensible default
plot	logical, whether to plot the effect of varname over time
...	other arguments to be passed to the plotting function.

Value

median, upper and lower confidence bands for the effect of varname over time; the funciton also produces a plot.

---

resetLoop *reset iterator*

Description

call this to reset an iterator's state to the initial

Usage

```
resetLoop(obj)
```

Arguments

obj	an mcmc iterator

---

residuals.mcmcspatsurv *resuiduals.mcmcspatsurv function*

Description

A function to compute Cox-Snell / modeified Cox-Snell / Martingale or Deviance residuals

Usage

```
## S3 method for class 'mcmcspatsurv'
residuals(object, type = "Cox-Snell", ...)
```

Arguments

object	an object produced by the function survspat
type	type of residuals to return. Possible choices are 'Cox-Snell', 'modified-Cox-Snell', 'Martingale' or 'deviance'.
...	other arguments (not used here)

Value

the residuals

---

setTxtProgressBar2 *set the progress bar*

Description

update a text progress bar. See help(txtProgressBar) for more info.

Usage

```
setTxtProgressBar2(pb, value, title = NULL, label = NULL)
```

Arguments

pb	text progress bar object
value	new value
title	ignored
label	text for end of progress bar

---

setupHazard *setupHazard function*

Description

A function to set up the baseline hazard, cumulative hazard and derivative functions for use in evaluating the log posterior. This fucntion is not intended for general use.

Usage

```
setupHazard(dist, pars, grad = FALSE, hess = FALSE)
```

Arguments

`dist`	an object of class 'basehazardspec'
`pars`	parameters with which to create the functions necessary to evaluate the log posterior
`grad`	logical, whetether to create gradient functions for the baseline hazard and cumulative hazard
`hess`	logical, whetether to create hessian functions for the baseline hazard and cumulative hazard

Value

a list of functions used in evaluating the log posterior

`setupPrecMatStruct` *setupPrecMatStruct function*

Description

A function to set up the computational grid and precision matrix structure for SPDE models.

Usage

```
setupPrecMatStruct(shape, cellwidth, no)
```

Arguments

`shape`	an object of class SpatialPolygons or SpatialPolygonsDataFrame
`cellwidth`	a scalar, the width of the grid cells
`no`	the order of the SPDE approximation: see Lindgren et al 2011 for details

Value

the computational grid and a function for constructing the precision matrix

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665
2. Finn Lindgren, Havard Rue, Johan Lindstrom. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B 73(4)

simsurv *simsurv function*

Description

A function to simulate spatial parametric proportional hazards model. The function works by simulating candidate survival times using MCMC in parallel for each individual based on each individual's covariates and the common parameter effects, beta.

Usage

```
simsurv(X = cbind(age = runif(100, 5, 50), sex = rbinom(100, 1, 0.5), cancer =
  rbinom(100, 1, 0.2)), beta = c(0.0296, 0.0261, 0.035), omega = 1,
  dist = "exp", coords = matrix(runif(2 * nrow(X)), nrow(X), 2),
  cov.parameters = c(1, 0.1), cov.model = covmodel(model = "exponential",
  pars = NULL), mcmc.control = mcmcpars(nits = 1e+05, burn = 10000, thin =
  90), savechains = TRUE)
```

Arguments

`X`	a matrix of covariate information
`beta`	the parameter effects
`omega`	vector of parameters for the baseline hazard model
`dist`	the distribution choice: exp or weibull at present
`coords`	matrix with 2 columns giving the coordinates at which to simulate data
`cov.parameters`	a vector: the parameters for the covariance function
`cov.model`	an object of class covmodel, see ?covmodel
`mcmc.control`	mcmc control paramters, see ?mcmcpars
`savechains`	save all chains? runs faster if set to FALSE, but then you'll be unable to conduct convergence/mixing diagnostics

Value

in list element 'survtimes', a vector of simulated survival times (the last simulated value from the MCMC chains) in list element 'T' the MCMC chains

See Also

covmodel, survspat, tpowHaz, exponentialHaz, gompertzHaz, makehamHaz, weibullHaz

spatialpars *spatialpars function*

Description

A function to return the mcmc chains for the spatial covariance function parameters

Usage

```
spatialpars(x)
```

Arguments

x	an object of class mcmcspatsurv

Value

the eta mcmc chains

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

spatsurvVignette *spatsurvVignette function*

Description

Display the introductory vignette for the spatsurv package.

Usage

```
spatsurvVignette()
```

Value

displays the vignette by calling browseURL

SPDE *SPDE function*

Description

A function to declare and evaluate an SPDE covariance function.

Usage

```
SPDE(ord)
```

Arguments

`ord`	the order of the model to be used, currently an integer between 1 an 3. See Lindgren 2011 paper.

Value

an covariance function based on the SPDE model

See Also

ExponentialCovFct, covmodel, CovarianceFct

SPDEprec *SPDEprec function*

Description

A function to used in entering elements into the precision matrix of an SPDE model. Not intended for general use.

Usage

```
SPDEprec(a, ord)
```

Arguments

`a`	parameter a, see Lindgren et al 2011.
`ord`	the order of the SPDE model, see Lindgren et al 2011.

Value

a function used for creating the precision matrix

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665
2. Finn Lindgren, Havard Rue, Johan Lindstrom. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B 73(4)

SpikedExponentialCovFct *SpikedExponentialCovFct function*

Description

A function to declare and also evaluate a spiked exponential covariance function. This is an exponential covariance function with a nugget.

Usage

```
SpikedExponentialCovFct()
```

Value

the spiked exponential covariance function

See Also

ExponentialCovFct, covmodel, CovarianceFct

---

spplot1 *spplot1 function*

Description

A function to provide spplot-like plotting capability but NOT using trellis graphics. This function also acts as an interface for fast plotting of SpatialPolygonsDataFrame or SpatialPixelsDataFrame objects using leaflet HTML plotting capabilities to get zoomable plots with real-world context: transformation to the correct projection is done automatically.

Usage

```
spplot1(x, what, palette = brewer.pal(5, "Oranges"), breaks = NULL,
  legpos = "topleft", fun = identity, include.lowest = TRUE, bty = "n",
  bg = NULL, printlegend = TRUE, bw = FALSE, useLeaflet = FALSE,
  urltemplate = urlTemplate("Stamen_Toner"), fillOpacity = 0.5,
  legendOpacity = 0.5, OSMbg = NULL, ...)
```

Arguments

x	a SpatialPolgonsDataFrame or a SpatialPointsDataFrame
what	the name of the variable to plot
palette	the palette, can either be a vector of names of colours, or a vector of colours produced for example by the brewer.pal function.
breaks	optional breaks for the legend, a vector of length 1 + length(palette)
legpos	the position of the legend, options are 'topleft', 'topright', 'bottomleft', 'bottomright'
fun	an optional function of the data to plot, default is the identity function
include.lowest	see ?cut
bty	see ?legend
bg	see ?legend
printlegend	logical: print the legend?
bw	Logical. Plot in black/white/greyscale? Default is to produce a colour plot. Useful for producing plots for journals that do not accept colour plots.
useLeaflet	whether to use leaflet to produce a zoomable map this requires the leaflet package, available by issuing the command "devtools::install_github('rstudio/leaflet')"
urltemplate	template for leaflet map background, default is urlTemplate('Stamen-Toner'), but any valid web address for leaflet templates will work here. See ?urlTemplate.
fillOpacity	see ?addPolygons
legendOpacity	see opacity argument in function addLegend
OSMbg	optional OpenStreetMap background to add to plot, obtain this using the function getBackground
...	other arguments to be passed to plot

Details

See http://leaflet-extras.github.io/leaflet-providers/preview/ for examples of leaflet templates.

Instructions on installing the leaflet R package are available from https://rstudio.github.io/leaflet/

Value

either produces a plot or if useLeaflet is TRUE, returns a leaflet map widget to which further layers can be added

See Also

urlTemplate, getBackground, brewer.pal

spplot_compare *spplot_compare function*

Description

A function to compare two SpatialPolgonsDataFrame or SpatialPointsDataFrame objects using a unified legend for the variable of interest in both

Usage

```
spplot_compare(x, y, what, what1 = what, palette = brewer.pal(9, "Oranges"),
  legpos = "topleft", border = NA, fun = identity, t1 = "", t2 = "",
  bw = FALSE, ...)
```

Arguments

x	a SpatialPolgonsDataFrame or a SpatialPointsDataFrame
y	a SpatialPolgonsDataFrame or a SpatialPointsDataFrame
what	the name of the variable from x to plot
what1	the name of the variable from y to plot. default is to plot the variable of the same name
palette	the palette, can either be a vector of names of colours, or a vector of colours produced for example by the brewer.pal function.
legpos	the position of the legend, options are 'topleft', 'topright', 'bottomleft', 'bottomright'
border	see ?spplot
fun	an optional function of the data to plot, default is the identity function
t1	title for the plot of x
t2	title for the plot of y
bw	Logical. Plot in black/white/greyscale? Default is to produce a colour plot. Useful for producing plots for journals that do not accept colour plots.
...	other arguments to be passed to the plot function

Value

produces a plot comparing x[[what]] and y[[what1]]

summary.mcmc *summary.mcmc function*

Description

summary of an mcmc iterator print out values of an iterator and reset it. DONT call this in a loop that uses this iterator - it will reset it. And break.

Usage

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

Arguments

object	an mcmc iterator
...	other args

summary.mcmcspatsurv *summary.mcmcspatsurv function*

Description

A function to return summary tables from an MCMC run

Usage

```
## S3 method for class 'mcmcspatsurv'
summary(object, probs = c(0.5, 0.025, 0.975), ...)
```

Arguments

object	an object inheriting class mcmcspatsurv
probs	vector of quantiles to return
...	additional arguments

Value

summary tables to the console

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, vcov.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance `surv3d` *Spatial Survival Plot in 3D*

Description

Do a 3d plot of spatial survival data

Usage

```
surv3d(spp, ss, lwd = 2, lcol = "black", lalpha = 1, pstyle = c("point",
  "text"), psize = c(20, 10), pcol = c("red", "black"), ptext = c("X",
  ""), palpha = 1, title = "Spatial Survival", basegrid = TRUE,
  baseplane = TRUE)
```

Arguments

`spp`	A spatial points data frame
`ss`	A Surv object (with right-censoring)
`lwd`	Line width for stems
`lcol`	Line colour for stems
`lalpha`	Opacity for stems
`pstyle`	Point style "point" or "text"
`psize`	Vector of length 2 for uncensored/censored points size
`pcol`	Vector of length 2 for uncensored/censored points colours
`ptext`	Vector of length 2 for uncensored/censored text characters
`palpha`	Opacity for points/text
`title`	Main title for plot
`basegrid`	add a grid at t=0
`baseplane`	add a plane at t=0

Details

Uses rgl graphics to make a spinny zoomy plot

Value

nothing

Author(s)

Barry S Rowlingson

Examples

```
## Not run:
require(sp)
require(survival)
d = data.frame(
  x=runif(40)*1.5,
  y = runif(40),
  age=as.integer(20+30*runif(40)),
  sex = sample(c("M","F"),40,TRUE)
)
coordinates(d)=~x+y
d$surv = Surv(as.integer(5+20*runif(40)),runif(40)>.9)
clear3d();surv3d(d,d$surv,baseplane=TRUE,basegrid=TRUE)
clear3d();surv3d(d,d$surv,baseplane=TRUE,basegrid=TRUE,pstyle="t",lalpha=0.5,lwd=3,palpha=1)

## End(Not run)
```

---

survival_PP *survival_PP function*

Description

A function to compute an individual's survival function

Usage

```
survival_PP(inputs)
```

Arguments

inputs	inputs for the function including the model matrix, frailties, fixed effects and the parameters of the baseline hazard derived from this model

Value

the survival function for the individual

---

survspat *survspat function*

Description

A function to run a Bayesian analysis on censored spatial survial data assuming a proportional hazards model using an adaptive Metropolis-adjusted Langevin algorithm.

Usage

```
survspat(formula, data, dist, cov.model, mcmc.control, priors, shape = NULL,
  ids = list(shpid = NULL, dataid = NULL),
  control = inference.control(gridded = FALSE))
```

Arguments

`formula`	the model formula in a format compatible with the function flexsurvreg from the flexsurv package
`data`	a SpatialPointsDataFrame object containing the survival data as one of the columns
`dist`	choice of distribution function for baseline hazard. Current options are: exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz
`cov.model`	an object of class covmodel, see ?covmodel ?ExponentialCovFct or ?SpikedExponentialCovFct
`mcmc.control`	mcmc control parameters, see ?mcmcpars
`priors`	an object of class Priors, see ?mcmcPriors
`shape`	when data is a data.frame, this can be a SpatialPolygonsDataFrame, or a SpatialPointsDataFrame, used to model spatial variation at the small region level. The regions are the polygons, or they represent the (possibly weighted) centroids of the polygons.
`ids`	named list entry shpid character string giving name of variable in shape to be matched to variable dataid in data. dataid is the second entry of the named list.
`control`	additional control parameters, see ?inference.control

Value

an object inheriting class 'mcmcspatsurv' for which there exist methods for printing, summarising and making inference from.

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

See Also

tpowHaz, exponentialHaz, gompertzHaz, makehamHaz, weibullHaz, covmodel, linkExponentialCovFct, `SpikedExponentialCovFct`, mcmcpars, mcmcPriors, inference.control `survspatNS` *survspatNS function*

Description

A function to perform maximun likelihood inference for non-spatial survival data.

Usage

```
survspatNS(formula, data, dist, control = inference.control())
```

Arguments

`formula`	the model formula in a format compatible with the function flexsurvreg from the flexsurv package
`data`	a SpatialPointsDataFrame object containing the survival data as one of the columns
`dist`	choice of distribution function for baseline hazard. Current options are: exponentialHaz, weibullHaz, gompertzHaz, makehamHaz, tpowHaz
`control`	additional control parameters, see ?inference.control

Value

an object inheriting class 'mcmcspatsurv' for which there exist methods for printing, summarising and making inference from.

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

See Also

tpowHaz, exponentialHaz, gompertzHaz, makehamHaz, weibullHaz, covmodel, linkExponentialCovFct, `SpikedExponentialCovFct`, mcmcpars, mcmcPriors, inference.control tpowHaz *tpowHaz function*

Description

A function to define a parametric proportional hazards model where the baseline hazard is taken from the 'powers of t' model. This function returns an object inheriting class 'basehazardspec', list of functions 'distinfo', 'basehazard', 'gradbasehazard', 'hessbasehazard', 'cumbasehazard', 'gradcumbasehazard', 'hesscumbasehazard' and 'densityquantile'

Usage

```
tpowHaz(powers)
```

Arguments

`powers`	a vector of powers of t. These are powers are treated as fixed in estimation routines and it is assumed that the log cumulatice baseline hazard is a linear combination of these powers of t

Details

The `distinfo` function is used to provide basic distribution specific information to other `spatsurv` functions. The user is required to provide the following information in the returned list: `npars`, the number of parameters in this distribution; `parnames`, the names of the parameters; `trans`, the transformation scale on which the priors will be provided; `itrans`, the inverse transformation function that will be applied to the parameters before the hazard, and other functions are evaluated; `jacobian`, the derivative of the inverse transformation function with respect to each of the parameters; and `hessian`, the second derivatives of the inverse transformation function with respect to each of the parameters – note that currently the package `spatsurv` only allows the use of functions where the parameters are transformed independently.

The `basehazard` function is used to evaluate the baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, `t` and returns a vector.

The `gradbasehazard` function is used to evaluate the gradient of the baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, `t`, and returns a matrix.

The `hessbasehazard` function is used to evaluate the Hessian of the baseline hazard function. It returns a function that accepts as input a vector of times, `t` and returns a list of hessian matrices corresponding to each `t`.

The `cumbasehazard` function is used to evaluate the cumulative baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, `t` and returns a vector.

The `gradcumbasehazard` function is used to evaluate the gradient of the cumulative baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, `t`, and returns a matrix.

The `hesscumbasehazard` function is used to evaluate the Hessian of the cumulative baseline hazard function. It returns a function that accepts as input a vector of times, `t` and returns a list of hessian matrices corresponding to each `t`.

The `densityquantile` function is used to return quantiles of the density function. This is NOT REQUIRED for running the MCMC, merely for us in post-processing with the `predict` function where `type` is 'densityquantile'. In the case of the Weibull model for the baseline hazard, it can be shown that the q-th quantile is:

Value

an object inheriting class 'basehazardspec'

See Also

exponentialHaz, gompertzHaz, makehamHaz, weibullHaz

`transformweibull` *transformweibull function*

Description

A function to back-transform estimates of the parameters of the weibull baseline hazard function, so they are commensurate with R's inbuilt density functions. Transforms from (shape, scale) to (alpha, lambda)

Usage

```
transformweibull(x)
```

Arguments

`x`	a vector of paramters

Value

the transformed parameters. For the weibull model, this is the back-transform from 'alpha' and 'lambda' to 'shape' 'scale' (see ?dweibull).

txtProgressBar2 *A text progress bar with label*

Description

This is the base txtProgressBar but with a little modification to implement the label parameter for style=3. For full info see txtProgressBar

Usage

```
txtProgressBar2(min = 0, max = 1, initial = 0, char = "=", width = NA,
  title = "", label = "", style = 1)
```

Arguments

min	min value for bar
max	max value for bar
initial	initial value for bar
char	the character (or character string) to form the progress bar.
width	progress bar width
title	ignored
label	text to put at the end of the bar
style	bar style

urlTemplate *urlTemplate function*

Description

A function to return a url for a leaflet template for use as map backgrounds with the spplot1 function.

Usage

```
urlTemplate(name = "Stamen_Toner")
```

Arguments

name	name of the template to use, the default is 'Stamen_Toner'

Details

Possible templates are: OpenStreetMap_Mapnik, OpenStreetMap_BlackAndWhite, OpenStreetMap_DE, OpenStreetMap_France, OpenStreetMap_HOT, OpenTopoMap, Thunderforest_OpenCycleMap, Thunderforest_Transport, Thunderforest_Landscape, Thunderforest_Outdoors, OpenMapSurfer_Roads, OpenMapSurfer_Grayscale, Hydda_Full, Hydda_Base, MapQuestOpen_OSM, Stamen_Toner, Stamen_TonerBackground, Stamen_TonerLite, Stamen_Watercolor, Stamen_Terrain, Stamen_TerrainBackground, Stamen_TopOSMRelief, Esri_WorldStreetMap, Esri_WorldTopoMap, Esri_WorldImagery, Esri_WorldTerrain, Esri_WorldShadedRelief, Esri_WorldPhysical, Esri_OceanBasemap, Esri_NatGeoWorldMap, Esri_WorldGrayCanvas, Acetate_all, Acetate_terrain, HERE_satelliteDay, HERE_hybridDayMobile, HERE_hybridDay

See `http://leaflet-extras.github.io/leaflet-providers/preview/` for other leaflet templates

Value

url for the leaflet template

`vcov.mcmcspatsurv` *vcov.mcmcspatsurv function*

Description

A function to return the variance covariance matrix of the parameters beta, omega and eta

Usage

```
## S3 method for class 'mcmcspatsurv'
vcov(object, ...)
```

Arguments

`object`	an object inheriting class mcmcspatsurv
`...`	other arguments, not used here

Value

the variance covariance matrix of the parameters beta, omega and eta

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

vcov.mlspatsurv *vcov.mlspatsurv function*

Description

A function to return the variance covariance matrix of the parameters beta, omega and eta

Usage

```
## S3 method for class 'mlspatsurv'
vcov(object, ...)
```

Arguments

object	an object inheriting class mcmcspatsurv
...	other arguments, not used here

Value

the variance covariance matrix of the parameters beta, omega and eta

See Also

print.mcmcspatsurv, quantile.mcmcspatsurv, summary.mcmcspatsurv, frailtylag1, spatialpars, hazardpars, fixedpars, randompars, baselinehazard, predict.mcmcspatsurv, priorposterior, posteriorcov, MCE, hazardexceedance

weibullHaz *weibullHaz function*

Description

A function to define a parametric proportional hazards model where the baseline hazard is taken from the Weibull model. This function returns an object inheriting class 'basehazardspec', list of functions 'distinfo', 'basehazard', 'gradbasehazard', 'hessbasehazard', 'cumbasehazard', 'gradcumbasehazard', 'hesscumbasehazard' and 'densityquantile'

Usage

```
weibullHaz()
```

Details

The distinfo function is used to provide basic distribution specific information to other spatsurv functions. The user is required to provide the following information in the returned list: npars, the number of parameters in this distribution; parnames, the names of the parameters; trans, the transformation scale on which the priors will be provided; itrans, the inverse transformation function that will be applied to the parameters before the hazard, and other functions are evaluated; jacobian, the derivative of the inverse transformation function with respect to each of the parameters; and hessian, the second derivatives of the inverse transformation function with respect to each of the parameters – note that currently the package spatsurv only allows the use of functions where the parameters are transformed independently.

The basehazard function is used to evaluate the baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, t and returns a vector.

The gradbasehazard function is used to evaluate the gradient of the baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, t, and returns a matrix.

The hessbasehazard function is used to evaluate the Hessian of the baseline hazard function. It returns a function that accepts as input a vector of times, t and returns a list of hessian matrices corresponding to each t.

The cumbasehazard function is used to evaluate the cumulative baseline hazard function for the distribution of interest. It returns a function that accepts as input a vector of times, t and returns a vector.

The gradcumbasehazard function is used to evaluate the gradient of the cumulative baseline hazard function with respect to the parameters, this typically returns a vector. It returns a function that accepts as input a vector of times, t, and returns a matrix.

The hesscumbasehazard function is used to evaluate the Hessian of the cumulative baseline hazard function. It returns a function that accepts as input a vector of times, t and returns a list of hessian matrices corresponding to each t.

The densityquantile function is used to return quantiles of the density function. This is NOT REQUIRED for running the MCMC, merely for us in post-processing with the predict function where type is 'densityquantile'. In the case of the Weibull model for the baseline hazard, it can be shown that the q-th quantile is:

Value

an object inheriting class 'basehazardspec'

See Also

tpowHaz, exponentialHaz, gompertzHaz, makehamHaz

`YfromGamma` *YfromGamma function*

Description

A function to change Gammas (white noise) into Ys (spatially correlated noise). Used in the MALA algorithm.

Usage

```
YfromGamma(Gamma, invrootQeigs, mu)
```

Arguments

`Gamma`	Gamma matrix
`invrootQeigs`	inverse square root of the eigenvectors of the precision matrix
`mu`	parameter of the latent Gaussian field

Value

Y

`YFromGamma_SPDE` *YFromGamma_SPDE function*

Description

A function to go from Gamma to Y

Usage

```
YFromGamma_SPDE(gamma, U, mu)
```

Arguments

`gamma`	Gamma
`U`	upper Cholesky matrix
`mu`	the mean

Value

the value of Y for the given Gamma

References

1. Benjamin M. Taylor. Auxiliary Variable Markov Chain Monte Carlo for Spatial Survival and Geostatistical Models. Benjamin M. Taylor. Submitted. http://arxiv.org/abs/1501.01665

2. Finn Lindgren, Havard Rue, Johan Lindstrom. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B 73(4)

Index