Package ‘ETAS.inlabru’

November 16, 2022

Type Package

Title This package uses inlabru to implement a Bayesian ETAS model for modelling seismic sequences

Version 0.1.0

Author Mark Naylor, Francesco Serafini, and Finn Lindgren

Maintainer The package maintainer `<mark.naylor@ed.ac.ukt>`

Description Modelling and inversion of ETAS model of seismicity using inlabru
The Epidemic Type Aftershock Sequence (ETAS) model is designed to model earthquakes that are triggered by previous events. In statistics, this is referred to as a Hawkes process.
The code can be used to generate synthetic ETAS catalogues which can also include some seeded events to model specific sequences.
We also implement a Bayesian inversion scheme using the Integrated Nested Laplace Approximation (INLA) using inlabru.
For the temporal model, given a training catalogue of times and magnitudes, the code returns the joint posteriors for all the ETAS parameters.
In the future roadmap, we will include tools to model the spatial distribution and spatio-temporal evolution of seismic sequences.

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 7.2.2

Roxygen list(markdown = TRUE)

Imports lemon, parallel, tidyquant, dplyr, ggplot2, foreach, INLA, inlabru

NeedsCompilation no

R topics documented:

`breaks_exp` *Title*

Description

Title

Usage

```
breaks_exp(tt_, T2_, coef_ = 2, delta_, N_exp_ = 10)
```

Arguments

`tt_`	List of the event times a grid is needed for days.
`T2_`	End of temporal domain days.
`coef_`	TimeBinning parameter:
`delta_`	TimeBinning parameter:
`N_exp_`	TimeBinning parameter: Number of bins

`compute.grid` *Title*

Description

Title

Usage

```
compute.grid(param., list.input_)
```

Arguments

`list.input_`

`create.input.list.temporal.noCatalogue` *Function to create a default input file for the ETAS Hawkes temporal model where no catalogue is specified in the input file*

Description

Function to create a default input file for the ETAS Hawkes temporal model where no catalogue is specified in the input file

Usage

```
create.input.list.temporal.noCatalogue(input_path)
```

Arguments

`input_path`	Input file and path as a string

Value

The formatted input.list with the elements required for the temporal Hawkes model

Examples

```
# HOW DO WE REFERENCE A FILE IN THE data DIRECTORY?
#create.input.list.temporal.noCatalogue('data/user_input_synthetic_noCatalog.txt')
```

create.input.list.temporal.withCatalogue

Function to create a default input file for the ETAS Hawkes temporal model where a catalogue is specified in the input file.

Description

Function to create a default input file for the ETAS Hawkes temporal model where a catalogue is specified in the input file.

Usage

```
create.input.list.temporal.withCatalogue(input_path)
```

Arguments

`input_path`

gamma.t

Gamma copula transformation: Conversion of ETAS para to internal scale

Description

Gamma copula transformation: Conversion of ETAS para to internal scale

Usage

```
## S3 method for class 't'
gamma(x, a, b)
```

Arguments

`b`

`generate.temporal.ETAS.synthetic` *Generates a sythetic catalogue using the ETAS model*

Description

Generates a sythetic catalogue using the ETAS model

Usage

```
generate.temporal.ETAS.synthetic(
  theta,
  beta.p,
  M0,
  T1,
  T2,
  Ht = NULL,
  ncore = 1
)
```

Arguments

`theta`	ETAS parameters `data.frame(mu=mu, K=K, alpha=alpha, c=c, p=p)`.
`beta.p`	Slope of GR relation: beta = b ln(10).
`M0`	The minimum magnitude in the synthetic catalogue.
`T1`	The start time for the synthetic catalogue days.
`T2`	The end time for the synthetic catalogue days.
`Ht`	A catalogue history to impose on the synthetic sequence.
`ncore`	Integer number of compute cores to use.

Value

A data.frame of the temporal catalogue with columns `[t_i, M_i, gen_i]` where, `t_i` are the times, `M_i` the magnitudes, `gen_i` includes information about the generation number

Examples

```
## EXAMPLE 1: Generate a 1000 day synthetic ETAS catalogue

generate.temporal.ETAS.synthetic( theta=data.frame(mu=0.1, K=0.089, alpha=2.29, c=0.11, p=1.


## EXAMPLE 2: To generate a 1000 day catalogue including a M6.7 event on day 500

Ht <- data.frame(ts=c(500), magnitudes=c(6.7))
generate.temporal.ETAS.synthetic( theta=data.frame(mu=0.1, K=0.089, alpha=2.29, c=0.11, p=1.
```

`get_posterior_N` *Title*

Description

Title

Usage

```
get_posterior_N(input.list)
```

Arguments

`input.list`

`get_posterior_param` *Generate summary information on the fitted ETAS model*

Description

Generate summary information on the fitted ETAS model

Usage

```
get_posterior_param(input.list)
```

Arguments

`input.list`	Which has combined the input file (for link functions) and bru output (for marginals)

Value

Data frame summary and summary plot

`gt` *Time triggering function - used by bayesianETAS Used for comparing output of inlabru with Bayesian ETAS MN: TODO Cross-reference the paper*

Description

Time triggering function - used by bayesianETAS Used for comparing output of inlabru with Bayesian ETAS MN: TODO Cross-reference the paper

Usage

```
gt(th, t, ti, mi, M0)
```

Arguments

`M0`	Minimum magnitude threshold

`gt.2` *Time triggering function - used by Inlabru MN: TODO Cross-reference the paper*

Description

Time triggering function - used by Inlabru MN: TODO Cross-reference the paper

Usage

```
gt.2(th, t, ti, mi, M0)
```

Arguments

`M0`	Minimum magnitude threshold

`Int.ETAS.time.trig.function` *Integrated ETAS time-triggering function*

Description

Integrated ETAS time-triggering function

Usage

```
Int.ETAS.time.trig.function(theta, th, T2)
```

Arguments

`theta`	ETAS parameters data.frame(mu=mu, K=K, alpha=alpha, c=c, p=p)
`th`	Time of past event? days
`T2`	End of temporal model domain.

`IntInjecIntensity` *Title*

Description

Title

Usage

```
IntInjecIntensity(a = 50, V.i = 1, tau = 10, T.i, T2)
```

Arguments

`a`	Event rate per unit volume injected
`V.i`	Injected volume
`tau`	Decau rate days
`T.i`	Time of injection event
`T2`	

`inv.exp.t` *Inverse exponential link function:*

Description

Inverse exponential link function:

Usage

```
inv.exp.t(x, rate)
```

Arguments

`rate`

`inv.gamma.t` *Inverse gamma copula transformation:*

Description

Inverse gamma copula transformation:

Usage

```
inv.gamma.t(x, a, b)
```

Arguments

`b`

`Inv.Int.ETAS.time.trig.function` *Inverse of integrated ETAS time-triggering function*

Description

Inverse of integrated ETAS time-triggering function

Usage

```
Inv.Int.ETAS.time.trig.function(theta, omega, th)
```

Arguments

`theta`	ETAS parameters data.frame(mu=mu, K=K, alpha=alpha, c=c, p=p)
`th`	

`Inv.IntInjecIntensity` *Title*

Description

Title

Usage

```
Inv.IntInjecIntensity(a = 50, V.i = 1, tau = 10, T.i, number.injected.events)
```

Arguments

`a`	Event rate per unit volume injected
`V.i`	Injected volume
`tau`	Decau rate days
`T.i`	Time of injection event
`number.injected.events`	

`inv.loggaus.t` *Inverse log-gaussian copula transformation:*

Description

Inverse log-gaussian copula transformation:

Usage

```
inv.loggaus.t(x, m, s)
```

Arguments

`s`

inv.unif.t *Inverse uniform copula transformation:*

Description

Inverse uniform copula transformation:

Usage

```
inv.unif.t(x, a, b)
```

Arguments

`b`

It_df *Title*

Description

Title

Usage

```
It_df(param_, time.df)
```

Arguments

`time.df`

lambda.N *Title*

Description

Title

Usage

```
lambda.N(th.mu, th.K, th.alpha, th.c, th.p, T1, T2, M0, Ht, link.functions)
```

Arguments

`link.functions`

lambda_ *Conditional intensity - used by bayesianETAS Used for comparing output of inlabru with Bayesian ETAS*

Description

Conditional intensity - used by bayesianETAS Used for comparing output of inlabru with Bayesian ETAS

Usage

```
lambda_(th, t, ti.v, mi.v, M0)
```

Arguments

M0	Minimum magnitude threshold

lambda_2 *conditional intensity (used by Inlabru)*

Description

conditional intensity (used by Inlabru)

Usage

```
lambda_2(th, t, ti.v, mi.v, M0)
```

Arguments

th	Set of trial ETAS parameters ??
M0	Minimum magnitude threshold

log.Lambda_h *integrated triggering function - used by bayesianETAS Used for comparing output of inlabru with Bayesian ETAS*

Description

integrated triggering function - used by bayesianETAS Used for comparing output of inlabru with Bayesian ETAS

Usage

```
## S3 method for class 'Lambda_h'
log(th, ti, mi, M0, T1, T2)
```

Arguments

M0	Minimum magnitude threshold
T1	Start of temporal model domain.
T2	End of temporal model domain.

log.Lambda_h2 *integrated triggering function - used by Inlabru ALSO USED IN GENERATION SAMPLES*

Description

integrated triggering function - used by Inlabru ALSO USED IN GENERATION SAMPLES

Usage

```
## S3 method for class 'Lambda_h2'
log(theta, ti, mi, M0, T1, T2)
```

Arguments

theta	ETAS parameters `data.frame(mu=mu, K=K, alpha=alpha, c=c, p=p)`.
ti	Time of parent event.
mi	Magnitude of parent event
M0	Minimum magnitude threshold
T1	Start of temporal model domain.
T2	End of temporal model domain.

loggaus.t *Log-gaussian copula transformation: Conversion of ETAS para to internal scale*

Description

Log-gaussian copula transformation: Conversion of ETAS para to internal scale

Usage

```
loggaus.t(x, m, s)
```

Arguments

`s`

omori_plot *Code to plot samples from the of posterior ETAS triggering function*

Description

Code to plot samples from the of posterior ETAS triggering function

Usage

```
omori_plot(list.input, n.samp = 10, t.end = 1, n.breaks = 100)
```

Arguments

`list.input`

`n.breaks`

Plot_grid *Title*

Description

Title

Usage

```
Plot_grid(
  xx = xx.,
  yy = yy.,
  delta_ = delta.,
  n.layer = n.layer.,
  bdy_ = bdy.,
  min.edge = min.edge.
)
```

Arguments

`min.edge`

post_sampling *Function to return a many (n.samp) samples from the posterior of the parameters*

Description

Function to return a many (n.samp) samples from the posterior of the parameters

Usage

```
post_sampling(input.list, n.samp)
```

Arguments

`input.list`	Which has combined the input file (for link functions) and bru output (for marginals)
`n.samp`	The number of samples to draw from the posteriors

Value

n.samp samples drawn from the posteriors.

`sample.GR.magnitudes` *Return a sample of magnitudes drawn from the GR distribution*

Description

Return a sample of magnitudes drawn from the GR distribution

Usage

```
sample.GR.magnitudes(n, beta.p, M0)
```

Arguments

`n`	Number of events in the sample.
`beta.p`	Related to the b-value via `b ln(10)`.
`M0`	Minimum magnitude for the sample.

Value

A list of magnitudes of length `n` drawn from a GR distribution.

Examples

```
sample.GR.magnitudes(n=100, beta.p=log(10), M0=2.5)
```

`sample.temoral.ETAS.daughters` *Generate a sample of new events `data.frame(t_i, M_i)` of length `n.ev` for one parent event occuring at time `t_h` using the ETAS model.*

Description

Generate a sample of new events `data.frame(t_i, M_i)` of length `n.ev` for one parent event occuring at time `t_h` using the ETAS model.

Usage

```
sample.temoral.ETAS.daughters(theta, beta.p, th, n.ev, M0, T1, T2)
```

Arguments

`theta`	ETAS parameters `data.frame(mu=mu, K=K, alpha=alpha, c=c, p=p)`.
`beta.p`	Slope of GR relation: beta = b ln(10).
`th`	Time of parent event days.
`n.ev`	The number of events to be placed.
`M0`	Minimum magnitude in synthetic catalogue.
`T1`	Start time for synthetic catalogue days.
`T2`	End time for synthetic catalogue days.

Value

Generate a sample of new events `data.frame(t_i, M_i)` from one parent

`sample.temoral.injection.events` *Title*

Description

Title

Usage

```
sample.temoral.injection.events(a = 50, V.i = 1, tau = 10, beta.p, M0, T.i, T2)
```

Arguments

`a`	Induced event rate per unit volume.
`V.i`	Injected volume
`tau`	Decay rate days.
`beta.p`	Related to the b-value via `b ln(10)`.
`M0`	Minimum magnitude threshold.
`T.i`	Time of injection days.
`T2`	End of temporal model domain days.

Value

Catalogue of parent events induced by injection data.frame(times, magnitudes)

sample.temporal.ETAS.generation

Take all previous parent events from `Ht=data.frame[ts, magnitudes]` *and generates their daughters events using the ETAS model*

Description

Take all previous parent events from `Ht=data.frame[ts, magnitudes]` and generates their daughters events using the ETAS model

Usage

```
sample.temporal.ETAS.generation(theta, beta.p, Ht, M0, T1, T2, ncore = 1)
```

Arguments

`theta`	ETAS parameters `data.frame(mu=mu, K=K, alpha=alpha, c=c, p=p).`
`beta.p`	Slope of GR relation: beta = b ln(10).
`Ht`	The set of parent events in the form `data.frame[ts, magnitudes]`
`M0`	The minimum earthquake magnitude in the synthetic catalogue.
`T1`	The start time for the synthetic catalogue days.
`T2`	The end time for the synthetic catalogue days.
`ncore`	The number of compute cores to use

Value

Return one generation of daughters from the parents in `Ht` in the form `data.frame(t_i, M_i).`

Examples

```
# The parents are specified in Ht
Ht <- data.frame(ts=c(500), magnitudes=c(6.7))
sample.temporal.ETAS.generation( theta=data.frame(mu=0.1, K=0.089, alpha=2.29, c=0.11, p=1.0
```

`sample.temporal.ETAS.times` *Sampling times for events triggered by a parent at th according to the ETAS triggering function*

Description

Sampling times for events triggered by a parent at th according to the ETAS triggering function

Usage

```
sample.temporal.ETAS.times(theta, n.ev, th, T2)
```

Arguments

`theta`	ETAS parameters `data.frame(mu=mu, K=K, alpha=alpha, c=c, p=p)`.
`n.ev`	Number of events to return in the sample in time domain (th, T2].
`th`	Time of the parent event producing n.ev daughters.
`T2`	End time of model domain.

Value

t.sample A list of times in the interval 0, T2 distributed according to the ETAS triggering function.

`temporal.ETAS` *Function to fit Hawkes process model*

Description

function to fit a temporal ETAS model using `inlabru`.

Usage

```
temporal.ETAS(
  sample.s,
  M0,
  T1,
  T2,
  link.functions = NULL,
  coef.t.,
  delta.t.,
  N.max.,
  bru.opt
)
```

Arguments

`sample.s` Observed events: `data.frame` with columns time (ts), magnitude (magnitudes), event identifier (idx.p). Column names must not be changed.

`M0` Minimum magnitude threshold, `scalar`

`T1` Start of temporal model domain, `scalar` measure unit of sample.s$ts.

`T2` End of temporal model domain, `scalar` measure unit of sample.s$ts.

`link.functions`
Functions to transform the parameters from the internal INLA scale to the ETAS scale. It must be a `list` of functions with names (mu, K, alpha, c_, p)

`coef.t.` TimeBinning parameter: parameter regulating the relative length of successive bins, `scalar`.

`delta.t.` TimeBinning parameter: parameter regulating the bins’ width, `scalar`.

`N.max.` TimeBinning parameter: parameter regulating the Number of bins (= `N.max` + 2), `scalar`.

`bru.opt` Runtime options for inlabru: See https://inlabru-org.github.io/inlabru/reference/bru_call_options.html, `list`

Value

The fitted model as a ’bru’ object, which is a list

`Temporal.ETAS.fit` *Fits the remporal ETAS model and returns the results. This function decomposes the input.list for the ‘Hawkes.bru2“ function.*

Description

Fits the remporal ETAS model and returns the results. This function decomposes the input.list for the ‘Hawkes.bru2“ function.

Usage

```
Temporal.ETAS.fit(input.list)
```

Arguments

`input.list` All input data and parameters are passed to inlabru via this structured list.

Value

The fitted model as a bru object, which is a list

time.grid *Generate a set of time bins for a specific event and return.*

Description

Generate a set of time bins for a specific event and return.

Usage

```
## S3 method for class 'grid'
time(data.point, coef.t, delta.t, T2., displaygrid = FALSE, N.exp.)
```

Arguments

`coef.t`	TimeBinning parameter:
`delta.t`	TimeBinning parameter:
`T2.`	End of the temporal domain days.
`displaygrid`	Boolean variable - whether to plot the grid
`N.exp.`	TimeBinning parameter:

Value

A set of time bins aggregated over all events MORE DETAIL

Examples

```
## EXAMPLE 1
events <- data.frame( ts=c(0,1 , 3 ), idx.p=c(1,2,3) )
T2 <- 20
N.exp <- 8
delta.t <- 0.1
coef.t <- 1
time.grid(events, coef.t, delta.t, T2, displaygrid = FALSE, N.exp)
```

unif.t *Uniform copula transformation: Conversion of ETAS para to internal scale*

Description

Uniform copula transformation: Conversion of ETAS para to internal scale

Usage

```
unif.t(x, a, b)
```

Arguments

`b`