Package ‘gfilmm’

October 13, 2022

Type Package

Title Generalized Fiducial Inference for Normal Linear Mixed Models

Version 2.0.5

Description Simulation of the generalized fiducial distribution for normal linear mixed models with interval data. Fiducial inference is somehow similar to Bayesian inference, in the sense that it is based on a distribution that represents the uncertainty about the parameters, like the posterior distribution in Bayesian statistics. It does not require a prior distribution, and it yields results close to frequentist results. Reference: Cisewski and Hannig (2012) <doi:10.1214/12-AOS1030>.

License GPL-3

URL https://github.com/stla/gfilmm

BugReports https://github.com/stla/gfilmm/issues

Depends R (>= 3.1.0)

Imports forcats, lazyeval, Matrix, parallel, Rcpp (>= 1.0.0), spatstat (>= 2.0.0), spatstat.geom, stats, utils

Suggests AOV1R, car, emmeans, GGally, kde1d, knitr, lmerTest, markdown, testthat

LinkingTo Rcpp, RcppEigen

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

SystemRequirements C++11

NeedsCompilation yes

Author Stéphane Laurent [aut, cre], Jessi Cisewski [aut, ctb] (author of the original Matlab code, <https://orcid.org/0000-0002-9656-2272>)

Maintainer Stéphane Laurent <laurent_step@outlook.fr>
gfiCDF

Description

Fiducial cumulative distribution function of a parameter of interest.

Usage

gfiCDF(parameter, gfi)

Arguments

parameter a right-sided formula defining the parameter of interest, like ~ sigma_error/(Intercept)
gfi a gfilm object (output of gfilm or gfilmPredictive)

Value

The fiducial cumulative distribution function of the parameter.

Examples

h <- 0.01
gfi <- gfilm(
  ~ cbind(yield-h, yield+h), ~ 1, ~ block, data = npk, N = 5000, nthreads = 2
)
F <- gfiCDF(~ sqrt(sigma_block^2 + sigma_error^2)/(Intercept), gfi)
plot(F, xlim = c(0, 0.3), main = "Coefficient of variation",
     ylab = expression("Pr("<="x")"))
F(0.2)
Fiducial confidence interval

**Description**

Fiducial confidence interval of a parameter of interest.

**Usage**

```r
gfiConfInt(parameter, gfi, conf = 0.95)
```

**Arguments**

- **parameter**: a right-sided formula defining the parameter of interest, like `~ sigma_error/(Intercept)`
- **gfi**: a `gfilmm` object (output of `gfilmm` or `gfilmmPredictive`)
- **conf**: confidence level

**Value**

The fiducial confidence interval of the parameter.

**Examples**

```r
h <- 0.01
gfi <- gfilmm(~ cbind(yield-h, yield+h), ~ 1, ~ block, data = npk, N = 5000, nthreads = 2)
gfiConfInt(~ sqrt(sigma_block^2 + sigma_error^2)/(Intercept), gfi)
```

---

**gfilmm**

*Generalized fiducial inference*

**Description**

Samples the fiducial distributions.

**Usage**

```r
gfilmm(
  y,
  fixed,
  random,
  data,
  N,
  thresh = N/2,
  long = FALSE,
)```

seed = NULL,
nthreads = parallel::detectCores()
)

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

Arguments

y
fixed
random
data
N
thresh
long
seed
nthreads
x
...

Value

A list with two components: a dataframe VERTEX, and a vector WEIGHT. It has class gfilmm.

References


Examples

h <- 0.01
gfi <- gfilmm(~ cbind(yield-h, yield+h), ~ 1, ~ block, data = npk, N = 5000, nthreads = 2)

# fiducial cumulative distribution function of the intercept:
Fintercept <- gfiCDF(~ \texttt{(Intercept)}, gfi)
plot(Fintercept, xlim = c(40, 65))

# fiducial confidence interval of the intercept:
gfiConfInt(~ \texttt{(Intercept)}, gfi)

# fiducial density function of the intercept:
library(kde1d)
kfit <- kde1d(gfi$VERTEX[, "(Intercept)"], weights = gfi$WEIGHT)
curve(dkde1d(x, kfit), from = 45, to = 65)
**gfilmmPredictive**

*Generalized fiducial predictive distributions*

**Description**
Simulations of the generalized fiducial predictive distributions.

**Usage**
gfilmmPredictive(gfi, newdata)

**Arguments**
- **gfi**: a gfilmm object
- **newdata**: dataframe in which to look for variables with which to predict, or NULL if the model is an intercept-only model without random effect

**Value**
A list with two fields: FPD, a dataframe containing the simulations, and WEIGHT, their weight. This is a gfilmm object.

**Note**
Actually the levels of the random effects given in newdata can be different from the original levels. For instance, in the example provided below, we enter `block = c("4", "6")`, but we could also enter `block = c("A", "B")`, even though "A" and "B" are not some levels of the block factor. Both options only mean that the two observations to predict are in two different blocks.

**Examples**
gfi <- gfilmm(~ cbind(yield-0.1, yield+0.1), ~ N, ~ block, npk, 2000, nthreads = 2)
fpd <- gfilmmPredictive(gfi, data.frame(N = c("0","1"), block = c("4","6")))
gfiSummary(fpd)

---

**gfiQuantile**

*Quantiles of a fiducial distribution*

**Description**
Quantiles of the fiducial distribution of a parameter of interest.

**Usage**
gfiQuantile(parameter, gfi, probs)
Arguments

parameter a right-sided formula defining the parameter of interest, like ~ sigma_error/(Intercept)`
gfi a `gfilmm` object (output of `gfilmm` or `gfilmmPredictive`) probs numeric vector of probabilities

Value

Numeric vector of quantiles, of the same length as `probs`.

Examples

```r
h <- 0.01
gfi <- gfilmm(~ cbind(yield-h, yield+h), ~ 1, ~ block, data = npk, N = 5000, nthreads = 2)
gfiQuantile(~ sqrt(sigma_block^2 + sigma_error^2), gfi, c(25, 50, 75)/100)
```

```r
data(KM41)
h <- 0.005
gfi <- gfilmm(~ cbind(y-h, y+h), ~ 1, ~ Batch, data = KM41, N = 5000, nthreads = 2)
gfiSummary(gfi)
```
**Description**

The dataset used in Krishnamoorthy & Mathew’s example 4.1.

**Usage**

```
data(KM41)
```

**Format**

A data frame with 25 rows and 2 columns.

**References**


**Examples**

```
data(KM41)
str(KM41)
table(KM41$Batch)
```

---

**pHdata**

*pH dataset*

**Description**

A dataset from ?? (I don’t remember).

**Usage**

```
data(pHdata)
```

**Format**

A data frame with 160 rows and 4 columns. Column SIRE is a factor nested in column DAM.

**Examples**

```
data(pHdata)
str(pHdata)
table(droplevels(pHdata[pHdata$DAM=="D1","SIRE"}))
table(droplevels(pHdata[pHdata$DAM=="D2","SIRE"]))
table(droplevels(pHdata[pHdata$DAM=="D3","SIRE")))
```
Index

* data
  KM41, 7
  pHdata, 7

  gfiCDF. 2
  gfiConfInt, 3
  gfilm, 2, 3, 3, 5, 6
  gfilmPredictive. 2, 3, 5, 6
  gfiQuantile, 5
  gfiSummary, 6

  KM41, 7

  pHdata, 7
  print.gfilm(gfilm), 3