Package ‘RLRsim’

March 25, 2020

Type Package
Title Exact (Restricted) Likelihood Ratio Tests for Mixed and Additive Models
Version 3.1-6
Maintainer Fabian Scheipl <fabian.scheipl@stat.uni-muenchen.de>
Description Rapid, simulation-based exact (restricted) likelihood ratio tests for testing the presence of variance components/nonparametric terms for models fit with nlme::lme(), lme4::lmer(), lmeTest::lmer(), gamm4::gamm4(), mgcv::gamm() and SemiPar::spm().
License GPL
URL https://github.com/fabian-s/RLRsim
BugReports https://github.com/fabian-s/RLRsim/issues
Depends R (>= 2.14.0)
Imports Rcpp (>= 0.11.1), lme4 (>= 1.1), mgcv, nlme
LinkingTo Rcpp
Enhances SemiPar, lmerTest
RoxygenNote 7.0.2
NeedsCompilation yes
Author Fabian Scheipl [aut, cre] (<https://orcid.org/0000-0001-8172-3603>), Ben Bolker [ctb] (<https://orcid.org/0000-0002-2127-0443>)
Repository CRAN
Date/Publication 2020-03-25 15:00:17 UTC

R topics documented:

RLRsim-package ................................................. 2
exactLRT ....................................................... 2
exactRLRT ..................................................... 4
extract.lmeDesign ............................................. 7
LRTSim ............................................................... 8

Index 11
**Description**

**RLRsim-package**  
*R* package for fast and exact (restricted) likelihood ratio tests for mixed and additive models.

**Description**

**RLRsim** implements fast simulation-based exact tests for variance components in mixed and additive models for conditionally Gaussian responses – i.e., tests for questions like:

- is the variance of my random intercept significantly different from 0?
- is this smooth effect significantly nonlinear?
- is this smooth effect significantly different from a constant effect?

The convenience functions **exactRLRT** and **exactLRT** can deal with fitted models from packages **lme4**, **nlme**, **gamm4**, **SemiPar** and **mgcv**’s **gamm**()-function. Workhorse functions **LRTSim** and **RLRTSim** accept design matrices as inputs directly and can thus be used more generally to generate exact critical values for the corresponding (restricted) likelihood ratio tests.

The theory behind these tests was first developed in:  

Power analyses and sensitivity studies for **RLRsim** can be found in:  

**Author(s)**

Fabian Scheipl (<fabian.scheipl@stat.uni-muenchen.de>), Ben Bolker

**exactLRT**  
*Likelihood Ratio Tests for simple linear mixed models*

**Description**

This function provides an exact likelihood ratio test based on simulated values from the finite sample distribution for simultaneous testing of the presence of the variance component and some restrictions of the fixed effects in a simple linear mixed model with known correlation structure of the random effect and i.i.d. errors.
exactLRT

Usage

```r
exactLRT(
  m,
  m0,
  seed = NA,
  nsim = 10000,
  log.grid.hi = 8,
  log.grid.lo = -10,
  gridlength = 200,
  parallel = c("no", "multicore", "snow"),
  ncpus = 1L,
  cl = NULL
)
```

Arguments

- **m**: The fitted model under the alternative; of class `lme`, `lmerMod` or `spm`
- **m0**: The fitted model under the null hypothesis; of class `lm`
- **seed**: Specify a seed for `set.seed`
- **nsim**: Number of values to simulate
- **log.grid.hi**: Lower value of the grid on the log scale. See `exactLRT`.
- **log.grid.lo**: Lower value of the grid on the log scale. See `exactLRT`.
- **gridlength**: Length of the grid. See `LRTSim`.
- **parallel**: The type of parallel operation to be used (if any). If missing, the default is "no parallelization".
- **ncpus**: integer: number of processes to be used in parallel operation: typically one would choose this to the number of available CPUs. Defaults to 1, i.e., no parallelization.
- **cl**: An optional parallel or snow cluster for use if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

The model under the alternative must be a linear mixed model \( y = X \beta + Zb + \varepsilon \) with a single random effect \( b \) with known correlation structure and error terms that are i.i.d. The hypothesis to be tested must be of the form

\[
H_0 : \beta_{p+1-q} = \beta_{p+1-q}^0, \ldots, \beta_p = \beta_p^0; \quad \text{Var}(b) = 0
\]

versus

\[
H_A : \beta_{p+1-q} \neq \beta_{p+1-q}^0 \text{ or } \ldots \\
\quad \text{or } \beta_p \neq \beta_p^0 \text{ or } \text{Var}(b) > 0
\]

We use the exact finite sample distribution of the likelihood ratio test statistic as derived by Crainiceanu & Ruppert (2004).
Value

A list of class htest containing the following components:

- statistic the observed likelihood ratio
- p p-value for the observed test statistic
- method a character string indicating what type of test was performed and how many values were simulated to determine the critical value
- sample the samples from the null distribution returned by LRTSim

Author(s)

Fabian Scheipl, updates for lme4.0-compatibility by Ben Bolker

References


See Also

LRTSim for the underlying simulation algorithm; RLRTSim and exactRLRT for restricted likelihood based tests

Examples

```r
library(nlme);
data(Orthodont);

# test for Sex:Age interaction and Subject-Intercept
mA<-lme(distance ~ Sex * I(age - 11), random = ~ 1 | Subject,
  data = Orthodont, method = "ML")
m0<-lm(distance ~ Sex + I(age - 11), data = Orthodont)
summary(mA)
summary(m0)
exactLRT(m = mA, m0 = m0)
```

Description

This function provides an (exact) restricted likelihood ratio test based on simulated values from the finite sample distribution for testing whether the variance of a random effect is 0 in a linear mixed model with known correlation structure of the tested random effect and i.i.d. errors.
Usage

```r
exactRLRT(
  m,
  mA = NULL,
  m0 = NULL,
  seed = NA,
  nsim = 10000,
  log.grid.hi = 8,
  log.grid.lo = -10,
  gridlength = 200,
  parallel = c("no", "multicore", "snow"),
  ncpus = 1L,
  cl = NULL
)
```

Arguments

- `m`: The fitted model under the alternative or, for testing in models with multiple variance components, the reduced model containing only the random effect to be tested (see Details), an `lme`, `lmerMod` or `spm` object.
- `mA`: The full model under the alternative for testing in models with multiple variance components.
- `m0`: The model under the null for testing in models with multiple variance components.
- `seed`: input for `set.seed`.
- `nsim`: Number of values to simulate.
- `log.grid.hi`: Lower value of the grid on the log scale. See `exactRLRT`.
- `log.grid.lo`: Lower value of the grid on the log scale. See `exactRLRT`.
- `gridlength`: Length of the grid. See `exactLRT`.
- `parallel`: The type of parallel operation to be used (if any). If missing, the default is "no parallelization".
- `ncpus`: integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs. Defaults to 1, i.e., no parallelization.
- `cl`: An optional parallel or snow cluster for use if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

Testing in models with only a single variance component require only the first argument `m`. For testing in models with multiple variance components, the fitted model `m` must contain only the random effect set to zero under the null hypothesis, while `mA` and `m0` are the models under the alternative and the null, respectively. For models with a single variance component, the simulated distribution is exact if the number of parameters (fixed and random) is smaller than the number of observations. Extensive simulation studies (see second reference below) confirm that the application
of the test to models with multiple variance components is safe and the simulated distribution is
correct as long as the number of parameters (fixed and random) is smaller than the number of
observations and the nuisance variance components are not superfluous or very small. We use the
finite sample distribution of the restricted likelihood ratio test statistic as derived by Crainiceanu &

No simulation is performed if the observed test statistic is 0. (i.e., if the fit of the model fitted under
the alternative is indistinguishable from the model fit under H0), since the p-value is always 1 in
this case.

Value
A list of class htest containing the following components:
A list of class htest containing the following components:

- statistic the observed likelihood ratio
- p p-value for the observed test statistic
- method a character string indicating what type of test was performed and how many values
  were simulated to determine the critical value
- sample the samples from the null distribution returned by RLRTSim

Author(s)
Fabian Scheipl, bug fixes by Andrzej Galecki, updates for lme4-compatibility by Ben Bolker

References

Testing for Zero Variance Components in Linear Mixed Models, *Journal of Computational and

variance or polynomial regression in additive and linear mixed models. *Computational Statistics &
Data Analysis*, 52(7):3283–3299.

See Also
RLRTSim for the underlying simulation algorithm; exactLRT for likelihood based tests

Examples

```r
data(sleepstudy, package = "lme4")
mA <- lme4::lmer(Reaction ~ I(Days-4.5) + (1|Subject) + (0 + I(Days-4.5)|Subject),
    data = sleepstudy)
m0 <- update(mA, . ~ - (0 + I(Days-4.5)|Subject))
m.slope <- update(mA, . ~ - (1|Subject))
#test for subject specific slopes:
```
extract.lmeDesign

extractRLRT(m.slope, mA, m0)

library(mgcv)
data(trees)
#test quadratic trend vs. smooth alternative
m.q<-gamm(I(log(Volume)) ~ Height + s(Girth, m = 3), data = trees,
method = "REML")$lme
exactRLRT(m.q)
#test linear trend vs. smooth alternative
m.l<-gamm(I(log(Volume)) ~ Height + s(Girth, m = 2), data = trees,
method = "REML")$lme
exactRLRT(m.l)

###

extract.lmeDesign

*Extract the Design of a linear mixed model*

#### Description

These functions extract various elements of the design of a fitted `lme`-, `mer` or `lmerMod`-Object. They are called by `exactRLRT` and `exactLRT`.

#### Usage

```r
eextract.lmeDesign(m)
```

#### Arguments

- `m` a fitted `lme`- or `merMod`-Object

#### Value

- `Vr` estimated covariance of the random effects divided by the estimated variance of the residuals
- `X` design of the fixed effects
- `Z` design of the random effects
- `sigmasq` variance of the residuals
- `lambda` ratios of the variances of the random effects and the variance of the residuals
- `y` response variable

#### Author(s)

Fabian Scheipl, `extract.lmerModDesign` by Ben Bolker. Many thanks to Andrzej Galecki and Tomasz Burzykowski for bug fixes.
Examples

```r
library(nlme)
design <- extract.lmeDesign(lme(distance ~ age + Sex, data = Orthodont,
                                random = ~ 1))
str(design)
```

LRTSim

**Simulation of the (Restricted) Likelihood Ratio Statistic**

Description

These functions simulate values from the (exact) finite sample distribution of the (restricted) likelihood ratio statistic for testing the presence of the variance component (and restrictions of the fixed effects) in a simple linear mixed model with known correlation structure of the random effect and i.i.d. errors. They are usually called by `exactLRT` or `exactRLRT`.

Usage

```r
LRTSim(
  X,  # The fixed effects design matrix of the model under the alternative
  Z,  # The random effects design matrix of the model under the alternative
  q,  # The number of parameters restrictions on the fixed effects (see Details)
  sqrt.Sigma,  # The upper triangular Cholesky factor of the correlation matrix of the random effect
  seed = NA,  # Specify a seed for set.seed
  nsim = 10000,  # Number of values to simulate
  log.grid.hi = 8,  # Lower value of the grid on the log scale. See Details
  log.grid.lo = -10,
  gridlength = 200,
  parallel = c("no", "multicore", "snow"),
  ncpus = 1L,
  cl = NULL
)
```

Arguments

- **X**: The fixed effects design matrix of the model under the alternative
- **Z**: The random effects design matrix of the model under the alternative
- **q**: The number of parameters restrictions on the fixed effects (see Details)
- **sqrt.Sigma**: The upper triangular Cholesky factor of the correlation matrix of the random effect
- **seed**: Specify a seed for `set.seed`
- **nsim**: Number of values to simulate
- **log.grid.hi**: Lower value of the grid on the log scale. See Details
log.grid.lo  Lower value of the grid on the log scale. See Details

gridlength  Length of the grid for the grid search over lambda. See Details

parallel  The type of parallel operation to be used (if any). If missing, the default is "no parallelization".

ncpus  integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs. Defaults to 1, i.e., no parallelization.

c1  An optional parallel or snow cluster for use if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.

Details

The model under the alternative must be a linear mixed model $y = X\beta + Zb + \epsilon$ with a single random effect $b$ with known correlation structure $\Sigma$ and i.i.d errors. The simulated distribution of the likelihood ratio statistic was derived by Crainiceanu & Ruppert (2004). The simulation algorithm uses a grid search over a log-regular grid of values of $\lambda = \frac{\text{Var}(b)}{\text{Var}(\epsilon)}$ to maximize the likelihood under the alternative for $\text{nsim}$ realizations of $y$ drawn under the null hypothesis. log.grid.hi and log.grid.lo are the lower and upper limits of this grid on the log scale. gridlength is the number of points on the grid. These are just wrapper functions for the underlying C code.

Value

A vector containing the the simulated values of the (R)LRT under the null, with attribute 'lambda' giving arg min($f(\lambda)$) (see Crainiceanu, Ruppert (2004)) for the simulations.

Author(s)

Fabian Scheipl; parallelization code adapted from boot package

References


See Also

exactLRT, exactRLRT for tests
Examples

```r
library(lme4)
g <- rep(1:10, e = 10)
x <- rnorm(100)
y <- 0.1 * x + rnorm(100)
m <- lmer(y ~ x + (1|g), REML=FALSE)
m0 <- lm(y ~ 1)

(obs.LRT <- 2*(logLik(m) - logLik(m0)))
X <- getME(m, "X")
Z <- t(as.matrix(getME(m, "Zt")))
sim.LRT <- LRTsim(X, Z, 1, diag(10))
(pval <- mean(sim.LRT > obs.LRT))
```
Index

+Topic **datagen**
  LRTSim, 8
+Topic **distribution**
  LRTSim, 8
+Topic **htest**
  exactLRT, 2
  exactRLRT, 4
+Topic **package**
  RLRsim-package, 2
+Topic **utilities**
  extract.lmeDesign, 7
  exactLRT, 2, 2, 3, 5, 6, 9
  exactRLRT, 2, 4, 4, 5, 9
  extract.lmeDesign, 7
  extract.imerModDesign
    (extract.lmeDesign), 7
LRTSim, 2–4, 8

RLRsim (RLRsim-package), 2
RLRsim-package, 2
RLRTSim, 2, 4, 6
RLRTSim (LRTSim), 8