Package ‘MMS’

February 19, 2015

Type Package
Title Fixed effects Selection in Linear Mixed Models
Version 3.00
Date 2014-07-04
Author F.Rohart
Maintainer Florian Rohart <florian.rohart@gmail.com>
Description Fixed effects Selection in Linear Mixed Models
License GPL-3
Depends glmnet,Matrix, mht (>= 3.00)
NeedsCompilation no
Repository CRAN
Date/Publication 2014-04-07 10:07:35

R topics documented:

MMS-package . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
lassop . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
lmme . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
mhtp . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
refit.mhtp . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

Index 12

<table>
<thead>
<tr>
<th>MMS-package</th>
<th>Fixed effects Selection in Linear Mixed Models</th>
</tr>
</thead>
</table>

Description

This package performs fixed effects selection in linear mixed models

Details
Two major functions:

**lassop** performs fixed effects selection in linear mixed models through a L1-penalization of the log-likelihood of the marginal model. The method optimizes a criterion via a multicycle ECM algorithm.

**mhtp** performs fixed effects selection in linear mixed models through a modification of the previous algorithm which allows the use of any variable selection method developed in linear models. It is a combination of the **mht** method from the mht-package and the modified algorithm.

**Author(s)**

Florian Rohart
Maintainer: florian.rohart@gmail.com

**References**

Multiple hypotheses testing for variable selection; F. Rohart
Selection of fixed effects in high dimensional linear mixed models using a multicycle ECM algorithm; F. Rohart & al

**Examples**

```r
## Not run:
N <- 20  # number of groups
p <- 20  # number of covariates (including intercept)
q <- 2   # number of random effect covariates
ni <- rep(6,N)  # observations per group
n <- sum(ni)  # total number of observations

grp <- factor(rep(1:N,ni))  # grouping variable
grp=rbind(grp,grp)

beta <- c(1,2,4,3,rep(0,p-3))  # fixed-effects coefficients
x <- cbind(1,matrix(rnorm(n*p),nrow=n))  # design matrix

u1=rnorm(N,0,sd=sqrt(2))
u2=rnorm(N,0,sd=sqrt(2))
b11 <- rep(u1,ni)
b12 <- rep(u2,ni)
b1 <- rbind(b11,b12)

z=x[,1:2,drop=FALSE]
```
### L1-penalization in linear mixed models

**Description**

Performs a L1-penalization in linear mixed models

**Usage**

```r
lassop(data, y, z, grp, D, mu, step, fix, rand, penalty.factor, alpha, showit)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>data</code></td>
<td>Input matrix of dimension n * p; each row is an observation vector. The intercept should be included in the first column as (1,...,1). If not, it is added.</td>
</tr>
<tr>
<td><code>y</code></td>
<td>Response variable of length n.</td>
</tr>
<tr>
<td><code>z</code></td>
<td>Random effects matrix. Of size n*q.</td>
</tr>
<tr>
<td><code>grp</code></td>
<td>Grouping variable of length n.</td>
</tr>
<tr>
<td><code>D</code></td>
<td>Logical value. If TRUE, the random effects are considered to be independent, i.e. ( \Psi ) is a diagonal matrix. ( D=\text{TRUE} ) should be used with nested grouping factors.</td>
</tr>
<tr>
<td><code>mu</code></td>
<td>Positive regularization number to be used for the Lasso.</td>
</tr>
<tr>
<td><code>step</code></td>
<td>The algorithm performs at most <code>step</code> iterations. Default is 3000.</td>
</tr>
<tr>
<td><code>fix</code></td>
<td>Number of variables which are not submitted to selection. They have to be in the first columns of data. Default is 1, the selection is not performed on the intercept.</td>
</tr>
<tr>
<td><code>rand</code></td>
<td>A vector of length q: each entry k is the position of the random effects number k in the data matrix, 0 otherwise. If z contains variables that have both a fixed and a random effect, it is advised to not submit them to selection.</td>
</tr>
<tr>
<td><code>penalty.factor</code></td>
<td>Argument of 'glmnet'. Separate penalty factors can be applied to each coefficient. This is a number that multiplies lambda to allow differential shrinkage. Can be 0 for some variables, which implies no shrinkage, and that variable is always included in the model. Default is 1 for all variables that are not in 1:fix.</td>
</tr>
</tbody>
</table>

```r
epsilon=rnorm(120)
y <- numeric(n)
for (k in 1:n) y[k] <- x[k,]*beta + t(z[k,])*bi[,k] + epsilon[k]

fit0=lmme(x[,1:4],y,z,grp)
fit1=lassop(x,y,z,grp, mu=0.2, fix=1, rand=c(1,2))
# fit1=lassop(x,y,z,grp, mu=0.2, fix=1, rand=c(1,2), showit=TRUE)

fit2=mhtp(x,y,z,grp, fix=1, rand=c(1,2), alpha=0.1, num=15)
# fit2=mhtp(x,y,z,grp, fix=1, rand=c(1,2), alpha=0.1, num=15, show=c(1,1,1))

## End(Not run)
alpha  Argument of 'glmnet'. The elasticnet mixing parameter, with $0 \leq \alpha \leq 1$.  
alpha=1 is the lasso penalty, and alpha=0 the ridge penalty.

showit Logical value. If TRUE, shows the iterations of the algorithm. Default is FALSE.

Details

This function performs fixed effects selection in linear mixed models through a L1-penalization of the log-likelihood of the marginal model. The method optimizes a criterion via a multicycle ECM algorithm at the regularization parameter mu.

Two algorithms are available: one when the random effects are assumed to be independent (D=TRUE) and one when they are not (D=FALSE). Selection on the random is only performed when D=TRUE.

Value

A `lassop` object is returned.

data  List of the user-data: the scaled matrix used in the algorithm, the first column being (1,...,1); Y and Z, which is the design matrix of the random effects.

beta  Estimation of the fixed effects.

fitted.values  Fitted values calculated with the fixed effects and the random effects.

Psi  Variance of the random effects. Matrix of dimension q*q.

sigma_e  Variance of the residuals.

it  Number of iterations of the algorithm.

converge Logical. TRUE if the algorithm has converged, FALSE otherwise.

u  Vector of the concatenation of the estimated random effects (u_1',...,u_q').

call  The call that produced this object.

mu  The penalty used in the algorithm.

Examples

```r
## not run:
N <- 20  # number of groups
p <- 80  # number of covariates (including intercept)
q <- 2  # number of random effect covariates
ni <- rep(6, N)  # observations per group
n <- sum(ni)  # total number of observations

gp <- factor(rep(1, N, ni))  # grouping variable
gp=rbind(gp, gp)

beta <- c(1, 2, 4, 3, rep(0, p-3))  # fixed-effects coefficients
x <- cbind(1, matrix(rnorm(n*p), nrow=n))  # design matrix

u1=rnorm(N,0,sd=sqrt(2))
u2=rnorm(N,0,sd=sqrt(2))
b11 <- rep(u1, ni)
b12 <- rep(u2, ni)
```
\begin{verbatim}
bi <- rbind(bi1, bi2)

d = x[,1:2, drop = FALSE]

epsilon = rnorm(120)
y <- numeric(n)
for (k in 1:n) y[k] <- x[k,] * beta + t(z[k,] * bi[,k]) + epsilon[k]

#### independent random effects
fit = lassop(x, y, z, grp, D=1, mu=0.2, fix=1, rand=c(1,2))

#### dependent random effects
fit = lassop(x, y, z, grp, mu=0.2, fix=1, rand=c(1,2))

## End (Not run)
\end{verbatim}

\begin{description}
\item[Description] Estimation of fixed and random effects in linear mixed models.
\item[Usage] lmme(data, y, z, grp, D, step, showit)
\item[Arguments]
\begin{description}
\item[data] Input matrix of dimension n * p; each row is an observation vector. The intercept should be included in the first column as (1, ..., 1). If not, it is added.
\item[y] Response variable of length n.
\item[z] Random effects matrix. Of size n * q.
\item[grp] Grouping variable of length n.
\item[D] Logical value. If TRUE, the random effects are considered to be independent, i.e. \( \Psi \) is a diagonal matrix. D=TRUE should be used with nested grouping factors.
\item[step] The algorithm performs at most step iterations. Default is 3000.
\item[showit] Logical value. If TRUE, shows the convergence process of the algorithm. Default is FALSE.
\end{description}
\item[Details] lmme performs an ML-estimation of fixed and random effects in linear mixed models when no selection is involved. Two algorithms are available: one when the random effects are assumed to be independent (D=TRUE) and one when they are not (D=FALSE).
\end{description}
Value

- data: List of the user-data: the scaled matrix used in the algorithm, the first column being (1,...,1); Y; z and grp.
- beta: Estimation of the selected fixed effects.
- psi: Variance of the random effects. Matrix of dimension q*q.
- sigma_e: Variance of the noise.
- fitted.values: Fitted values calculated with the fixed effects and the random effects.
- it: Number of iterations of the algorithm.
- converge: Did the algorithm converge?
- u: Vector of the concatenation of the estimated random effects (u_1',...,u_q')'.
- call: The call that produced this object.

Examples

```r
## Not run:
N <- 20        # number of groups
p <- 80        # number of covariates (including intercept)
q <- 2         # number of random effect covariates
ni <- rep(6,N) # observations per group
n <- sum(ni)   # total number of observations

grp <- factor(rep(1:N,ni)) # grouping variable
gr=bind(grp,grp)

beta <- c(1,2,4,3,rep(0,p-3)) # fixed-effects coefficients
x <- cbind(1,matrix(rnorm(n*p),nrow=n)) # design matrix
u1=rnorm(N,0,sd=sqrt(2))
u2=rnorm(N,0,sd=sqrt(2))
bi1 <- rep(u1,ni)
bi2 <- rep(u2,ni)
bi <- rbind(bi1,bi2)

z=x[,1:2,drop=FALSE]

epsilon=rnorm(120)
y <- numeric(n)
for (k in 1:n) y[k] <- x[k,]%*%beta + t(z[k,])%*%bi[,k] + epsilon[k]

####
fit=lmme(x,y,z,grp)

## End(Not run)
```
**mhtp**

*Multiple testing procedure for variable selection in linear mixed models*

**Description**

Performs a multiple hypotheses testing in linear mixed models

**Usage**

```r
mhtp(data,Y,z,grp,D,fix,rand,alpha,step,num,ordre,m,show,IT,maxq,speed)
```

**Arguments**

- `data` Input matrix of dimension n * p; each row is an observation vector. The intercept should be included in the first column as (1,...,1). If not, it is added.
- `Y` Response variable of length n.
- `z` Random effects matrix. Of size n*q.
- `grp` Grouping variable of length n.
- `D` Logical value. If TRUE, the random effects are considered to be independent, i.e. \( \Psi \) is a diagonal matrix. D=TRUE should be used with nested grouping factors.
- `fix` Number of variables which are not submitted to selection. They have to be in the first columns of data. Default is 1, the selection is not performed on the intercept.
- `rand` A vector of length q: each entry k is the position of the random effects number k in the data matrix, 0 otherwise. If z contains variables that have both a fixed and a random effect, it is advised to not submit them to selection.
- `alpha` A user supplied type I error sequence. Default is (0.1,0.05).
- `step` The algorithm performs at most step iterations. Default is 3000.
- `num` Number of variables one wishes to order. Default is min(n-1,p-1,30).
- `ordre` Several possible algorithms to order the variables, ordre=c("bolasso","pval","pval_hd","FR"). "bolasso" uses the dyadic algorithm with the Bolasso technique, "pval" uses the p-values obtained with a regression on the full set of variables (only when p<n), "pval_hd" uses marginal regression, "FR" uses Forward Regression. Default is "bolasso".
- `m` Number of bootstrapped iteration of the Lasso. Only use if the algorithm is set to "bolasso". Default is m=10.
- `show` Vector of logical values, show=(showordre,showresult,showit). Default is (0,0,0). If showordre=TRUE, show the ordered variables at each step of the algorithm. if showresult=TRUE, show the value of the statistics and the estimated quantile at each step of the procedure; if onde=bolasso. if showit=TRUE, show the iterations of the algorithm.
mhtp

Number of simulations in the calculation of the quantile. Default is 10000.

maxq

Number of maximum multiple hypotheses testing to do. Default is \( \min(\log(\min(n,p)-1,2),5) \).

speed

Logical value. If TRUE, the algorithm is speeded up once the criteria convergence in beta and u are fulfilled.

Details

mhtp performs fixed effects selection in linear mixed models. It is a combination of the mht function from the mht-package with the algorithm used in the lassop function; so you might want to have a look at the help of mht.

Value

A 'mhtp' object is returned for which refit is available.

data

List of the user-data: the scaled matrix used in the algorithm, the first column being \((1,\ldots,1);Y;z\) and grp.

beta

The estimated vector of fixed effects coefficients. Each row concern a specific user level alpha.

fitted.values

Fitted values calculated with the fixed effects and the random effects.

u

Matrix with \#alpha columns. Each column is the concatenation of the estimated random effects \((u_1',\ldots,u_q')\)' for the user level alpha.

Psi

Variance of the random effects. Matrix of dimension q*q.

sigma_e

Variance of the residuals.

it

Number of iterations of the algorithm.

quantile

Array of all the estimated quantiles calculated during the procedure.

ordrebeta

All different order that has been used during the procedure.

converge

Did the algorithm converge?

call

The call that produced this object.

arg

List of all the arguments of the function (used to refit the function).

See Also

mht, refit.mhtp

Examples

```r
## Not run:
N <- 20 # number of groups
p <- 20 # number of covariates (including intercept)
q <- 2 # number of random effect covariates
ni <- rep(6,N) # observations per group
n <- sum(ni) # total number of observations
grp <- factor(rep(1:N,ni)) # grouping variable
```
refit.mhtp

Multiple testing procedure for variable selection in linear mixed models

**Description**
Performs a multiple hypotheses testing in linear mixed models

**Usage**

```r
## S3 method for class 'mhtp'
refit(object, Ynew, z, grp, D, fix, rand, alpha, step, num, order, m, show, IT, maxq, speed, ...)
```

**Arguments**

- `object` Object of class "mhtp".
- `Ynew` Response variable of length n.
- `z` Random effects matrix. Of size n*q.
- `grp` Grouping variable of length n.
- `D` Logical value. If TRUE, the random effects are considered to be independent, i.e. \( \Psi \) is a diagonal matrix. \( D=TRUE \) should be used with nested grouping factors.
- `fix` Number of variables which are not submitted to selection. They have to be in the first columns of data. Default is 1, the selection is not performed on the intercept.

---

```r
grp=rbind(grp,grp)

beta <- c(1,2,4,3,rep(0,p-3)) # fixed-effects coefficients
x <- cbind(1,matrix(rnorm(n*p),ncol=n)) # design matrix

u1=rnorm(N,0,sd=sqrt(2))
u2=rnorm(N,0,sd=sqrt(2))
b1 <- rep(u1,n1)
b2 <- rep(u2,n1)
b <- rbind(b1,b2)

z=x[,1:2,drop=FALSE]

epsilon=rnorm(120)
y <- numeric(n)
for (k in 1:n) y[k] <- x[k,]%*%beta + t(z[k,])%*%bi[,k] + epsilon[k]

fit=mhtp(x,y,z,grp,D=0,fix=1,rand=c(1,2),alpha=0.1,num=15)
#fit=mhtp(x,y,z,grp,D=0,fix=1,rand=c(1,2),alpha=0.1,num=15,show=c(1,1,1))
```

---

**refit.mhtp**

Multiple testing procedure for variable selection in linear mixed models
**rand**  
A vector of length q: each entry k is the position of the random effects number k in the data matrix, 0 otherwise. If z contains variables that have both a fixed and a random effect, it is advised to not submit them to selection.

**alpha**  
A user supplied type I error sequence. Default is (0.1,0.05).

**step**  
The algorithm performs at most step iterations. Default is 3000.

**num**  
Number of variables one wishes to order. Default is min(n-1,p-1,30).

**ordre**  
Several possible algorithms to order the variables, ordre=c("bolasso","pval","pval_hd","FR").  
"bolasso" uses the dyadic algorithm with the Bolasso technique, "pval" uses the p-values obtained with a regression on the full set of variables (only when p<n),  
"pval_hd" uses marginal regression, "FR" uses Forward Regression. Default is "bolasso".

**m**  
Number of bootstrapped iteration of the Lasso. Only use if the algorithm is set to "bolasso". Default is m=10.

**show**  
Vector of logical values, show=(showordre,showresult,showit). Default is (0,0,0).  
If showordre==TRUE, show the ordered variables at each step of the algorithm.  
if showresult==TRUE, show the value of the statistics and the estimated quantile at each step of the procedure; if ordre=bolasso. if showit==TRUE, show the iterations of the algorithm.

**IT**  
Number of simulations in the calculation of the quantile. Default is 10000.

**maxq**  
Number of maximum multiple hypotheses testing to do. Default is min(log(min(n,p)-1,2),5).

**speed**  
Logical value. If TRUE, the algorithm is speeded up once the criterion convergence in beta and u is fulfilled.

...  
not used

**Details**

See mhtp for more details.

**Value**

A＇mhtp object＇ is returned.

**data**  
List of the user-data: the scaled matrix used in the algorithm, the first column being (1,...,1); Y; z and grp.

**beta**  
Estimation of the selected fixed effects.

**fitted.values**  
Fitted values calculated with the fixed effects and the random effects.

**u**  
Matrix with #alpha columns. Each column is the concatenation of the estimated random effects (u_1',...,u_q') for the user level alpha.

**Psi**  
Variance of the random effects. Matrix of dimension q*q.

**sigma_e**  
Variance of the noise.

**it**  
Number of iterations of the algorithm.

**quantile**  
Array of all the estimated quantiles calculated during the procedure.

**ordrebeta**  
All different order that has been used during the procedure.
converge Did the algorithm converge?
call The call that produced this object.
arg List of all the arguments of the function.

See Also
mhtp

Examples

```r
## Not run:
N <- 20 # number of groups
p <- 20 # number of covariates (including intercept)
q <- 2 # number of random effect covariates
ni <- rep(6,N) # observations per group
n <- sum(ni) # total number of observations

grp <- factor(rep(1:N,ni)) # grouping variable
grp=rbind(grp,grp)

beta <- c(1,2,4,3,rep(0,p-3)) # fixed-effects coefficients
x <- cbind(1,matrix(rnorm(n*p),nrow=n)) # design matrix

u1=rnorm(N,0,sd=sqrt(2))
u2=rnorm(N,0,sd=sqrt(2))
b1 <- rep(u1,ni)
b2 <- rep(u2,ni)
bi <- rbind(b1,b2)

z=x[,1:2,drop=FALSE]

epsilon=rnorm(120)
y <- numeric(n)
for (k in 1:n) y[k] <- x[k,]%*%beta + t(z[k,])%*%bi[,k] + epsilon[k]

#########
fit=mhtp(x,y,z,grp,D=0,fix=1,rand=c(1,2),alpha=0.1,num=15)
fit2=refit(fit,Ynew=y)

## End(Not run)
```
Index

lassop, 2, 3, 8
lmme, 5

mht, 2, 8
mht-package, 8
mhtp, 2, 7, 10, 11
MMS-package, 1

refit.mhtp, 8, 9