Package ‘lmmlasso’

February 20, 2015

Type Package
Title Linear mixed-effects models with Lasso
Version 0.1-2
Date 2011-04-13
Author Juerg Schelldorfer
Maintainer Juerg Schelldorfer <schell@stat.math.ethz.ch>
Description This package fits (gaussian) linear mixed-effects models for high-dimensional data (n<<p) using a Lasso-type approach for the fixed-effects parameter.
Depends methods, emulator, miscTools, penalized
License GPL
LazyLoad yes
Repository CRAN
Repository/R-Forge/Project lmmlasso
Repository/R-Forge/Revision 5
Date/Publication 2011-04-15 10:29:50
NeedsCompilation no

R topics documented:

lmmlasso-package .................................................. 2
classroomStudy .................................................. 2
lmmlasso .......................................................... 3
lmmlassoControl .................................................. 7
plot.lmmlasso .................................................. 8
print.lmmlasso .................................................. 9
summary.lmmlasso .............................................. 9

Index 11

1
Description

Fits (gaussian) linear mixed-effects models with lasso penalty for the fixed effects.

Details

Package: lmmlasso
Type: Package
Version: 0.1-2
Date: 2010-08-19
License: GPL
LazyLoad: yes

This is an early test version.

Author(s)

Juerg Schelldorfer
Maintainer: Juerg Schelldorfer <schell@stat.math.ethz.ch>

References


Description

This is a subset of the Classroom Study described in West et. al. (2007) with 156 observations and 6 variables.

Usage

data(classroomStudy)
Format
A list with the following four components.

Response variable. The students math achievement gain.

group Grouping variable comprising the class of the students.

X Fixed-effect design matrix. The first column is the intercept, then sex, minority, mathkind, ses, yearstea and mathprep.

Z Random-effects design matrix. Only one column in order to fit a random intercept model.

Details
A detailed description of the covariates can be found in West et. al. (2007) and was carried out by Hill et. al. (2005)

Source

http://www-personal.umich.edu/~bwest/classroom.csv

References


Examples

data(classroomStudy)

```r
lmmlasso(x, y, z = x, grp, weights = NULL, coefInit=NULL, lambda, startValue = 1, nonpen = 1:dim(z)[[2]], pdMat = c("pdIdent","pdDiag","pdSym"), method = "ML", CovOpt = c("nlminb","optimize"), stopSat = TRUE, standardize = TRUE, control = lmmlassoControl(), ranInd=1:dim(z)[[2]],...)
```

Description
Fits the solution for a high-dimensional (gaussian) linear mixed-effects models

Usage

`lmmlasso(x, ...)`

## Default S3 method:
lmmlasso(x, y, z = x, grp, weights = NULL, coefInit=NULL, lambda, startValue = 1, nonpen = 1:dim(z)[[2]], pdMat = c("pdIdent","pdDiag","pdSym"), method = "ML", CovOpt = c("nlminb","optimize"), stopSat = TRUE, standardize = TRUE, control = lmmlassoControl(), ranInd=1:dim(z)[[2]],...)
Arguments

- **x**: matrix of dimension ntot x p including the fixed-effects covariates. An intercept has to be included in the first column as (1,...,1).
- **y**: response variable of length ntot.
- **z**: random effects matrix of dimension ntot x q. It has to be a matrix, even if q=1.
- **grp**: grouping variable of length ntot
- **weights**: weights for the fixed-effects covariates: NA means no penalization, 0 means drop this covariate; if given, the argument nonpen is ignored. By default each covariate has weight 1
- **coefInit**: list with three components used as starting values for the fixed effects, the random effects variance components and the error standard deviation.
- **lambda**: positive regularization parameter
- **startValue**: Choice of the starting values for the fixed effects using linear regression. 1 means 10-fold cross-validation with L1-penalty, 2 means 10-fold cross-validation Ridge Regression and 0 means that all the covariates are set to zero and the intercept is the mean of the response variable
- **nonpen**: index of fixed effects with no penalization, ignored if the argument weights is specified, default is 1, which means that only the intercept (the first column in X) is not penalized.
- **pdMat**: Covariance structure for the random effects. pdIdent, pdDiag and pdSym are already implemented. Default to pdIdent. pdIdent: \( b_i \sim N(0, \theta^2 I) \) (1 parameter), pdDiag: \( b_i \sim N(0, diag(\theta_1, \ldots, \theta_q)) \) (q parameters), pdSym: \( b_i \sim N(0, \Psi) \) where \( \Psi \) is symmetric positive definite (\( q(q+1)/2 \) parameters)
- **method**: Only "ML" is allowed. "REML" is not yet implemented.
- **CovOpt**: which optimization routine should be used for updating the variance components parameters. optimize or nlminb. nlminb uses the estimate of the last iteration as a starting value. nlminb is faster if there are many Gauss-Seidel iterations.
- **stopSat**: logical. Should the algorithm stop when ntot > p?
- **standardize**: Should the x matrix be standardized such that each column has mean 0 and standard deviation 1? Be careful if the x matrix includes dummy variables.
- **control**: control parameters for the algorithm and the Armijo Rule, see lmmlassoControl for the details
- **ranInd**: Index of the random effects with respect to the x matrix
  ...

Details

All the details of the algorithm can be found in Schelldorfer et. al. (2010).

Value

A `lmmlasso` object is returned, for which `coef, resid, fitted, print, summary, plot` methods exist.
coefficients: estimated fixed-effects coefficients \( \hat{\beta} \)

pars: free parameters in the covariance matrix \( \Psi \) of the random effects

sigma: standard deviation \( \hat{\sigma} \) of the errors

random: vector with random effects, sorted by groups

u: vector with the standardized random effects, sorted by effect

ranef: vector with random effects, sorted by effect

fixef: estimated fixed-effects coefficients \( \hat{\beta} \)

fitted.values: The fitted values \( \hat{y} = \hat{X}\hat{\beta} + \hat{Z}\hat{b}_i \)

residuals: raw residuals \( y - \hat{y} \)

Psi: Covariance matrix \( \Psi \) of the random effects

corrPsi: Correlation matrix of the random effects

logLik: value of the log-likelihood function

deviance: \( \text{deviance} = -2 \cdot \log \text{Lik} \)

npar: Number of parameters. Corresponds to the cardinality of the active set of coefficients plus the number of free parameters in \( \Psi \)

aic: AIC

bic: BIC

data: data set, as a list with four components: x, y, z, grp (see above)

weights: weights for the fixed-effects covariates

nonpen: nonpenalized covariates. Differ from the input if weights is explicitly given

coeffInit: list with three components used as starting values

lambda: positive regularization parameter

converged: Does the algorithm converge? 0: correct convergence; an odd number means that maxIter was reached; an even number means that the Armijo step was not successful. For each unsuccessful Armijo step, 2 is added to converged. If converged is large compared to the number of iterations counter, you may increase maxArmijo.

counter: The number of iterations used.

stopped: logical. Does the algorithm stopped due to ntot > p?

pdMat: Covariance structure for the random effects

method: "ML"

CovOpt: optimization routine

control: see lmmlassoControl

call: call

stopped: logical. Does the algorithm stopped due to a too large active set?

ranInd: Index of the random effects with respect to the x matrix

objective: Value of the objective function at the final estimates
Author(s)
Juerg Schelldorfer, <schell@stat.math.ethz.ch>

References

Examples

```
# (1) Use the lmmlasso on the classroomStudy data set
data(classroomStudy)
fit1 <- lmmlasso(x=classroomStudy$x,y=classroomStudy$y,z=classroomStudy$z,
grp=classroomStudy$grp,lambda=15,pdMat="pdIdent")
summary(fit1)
plot(fit1)

# (2) Use the lmmlasso on a small simulated data set
set.seed(54)
N <- 20       # number of groups
p <- 6        # number of covariates (including intercept)
q <- 2        # number of random effect covariates
ni <- rep(6,N) # observations per group
ntot <- sum(ni) # total number of observations

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

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

bi1 <- rep(rnorm(N,0,3),each=ni) # Psi=diag(3,2)
bi2 <- rep(rnorm(N,0,2),each=ni)
bi <- rbind(bi1,bi2)

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

y <- numeric(ntot)
for (k in 1:ntot) y[k] <- x[k,]*beta + t(z[k,])%*%bi[,grp[k]] + rnorm(1)

# correct random effects structure
fit2 <- lmmlasso(x=x,y=y,z,z,grp=grp,lambda=10,pdMat="pdDiag")
summary(fit2)
plot(fit2)

# wrong random effects structure
fit3 <- lmmlasso(x=x,y=y,z=x[,1:3],grp=grp,lambda=10,pdMat="pdDiag")
summary(fit3)
plot(fit3)
```
lmlassoControl

Options for the lmlasso Algorithm

Description

Definition of various kinds of options in the algorithm.

Usage

```r
lmlassoControl(tol = 10^(-4), trace = 1, maxIter = 1000,
               maxArmijo = 20, number = 5, a_init = 1, delta = 0.1, rho = 0.001,
               gamma = 0, lower = 10^(-6), upper = 10^8, seed = 532,
               VarInt = c(0, 10), CovInt = c(-5, 5), thres = 10^(-4))
```

Arguments

tol convergence tolerance
trace integer. 1 prints no output, 2 prints warnings, 3 prints the current function values and warnings (not recommended)
maxIter maximum number of (outer) iterations
maxArmijo maximum number of steps to be chosen in the Armijo Rule. If the maximum is reached, the algorithm continues with optimizing the next coordinate.
number integer. Determines the active set algorithm. The zero fixed-effects coefficients are only updated each number iteration. It may be that a smaller number increases the speed of the algorithm. Use 0 ≤ number ≤ 5.
a_init $\alpha_{init}$ in the Armijo step. See Schelldorfer et al. (2010).
delta $\delta$ in the Armijo step. See Schelldorfer et al. (2010)
rho $\rho$ in the Armijo step. See Schelldorfer et al. (2010)
gamma $\gamma$ in the Armijo step. See Schelldorfer et al. (2010)
lower lower bound for the Hessian
upper upper bound for the Hessian
seed set.seed for calculating the starting value, which performs a 10-fold cross-validation.
VarInt Only for opt="optimize". The interval for the variance parameters used in "optimize". See help("optimize")
CovInt Only for opt="optimize". The interval for the covariance parameters used in "optimize". See help("optimize")
thres If a variance or covariance parameter has smaller absolute value than thres, the parameter is set to exactly zero.

Details

For the Armijo step parameters, see Bertsekas (2003)
**References**


---

**plot.lmlasso**

*Diagnostic Plots for a lmlasso object*

**Description**

Plots six diagnostic plots for checking the model assumptions and supporting model selection for a `lmlasso` object.

**Usage**

```r
## S3 method for class 'lmlasso'
plot(x, ...)```

**Arguments**

- `x`: a `lmlasso` object
- `...`: not used.

**Details**

`plot.lmlasso` shows six diagnostic plots which support checking the model assumption, model fit and may give hints for another model. 1) The first plot depicts the Tukey-Anscombe plot in order to check the assumptions about the errors. Points with the same color belong to the same group. 2) QQ-Plot of the residuals with equal coloring for each group. 3) QQ-Plot of the predicted random effects. Be careful with the interpretation since the random effects have not been standardized. The color shows which points belong to the same random-effects covariate. 4) Boxplot of the predicted random effects for each random-effects variable. 5) Plot of the fitted y-values against the observed y-values. 6) A histogram of the fixed-effects coefficients.

**Examples**

```r
data(classroomStudy)
fit <- lmlasso(x=classroomStudy$X, y=classroomStudy$y, z=classroomStudy$Z, grp=classroomStudy$grp, lambda=15, pdMat)
plot(fit)
```
print.lmlasso

Print a short summary of an \texttt{lmlasso} object.

Description

Prints a short summary of an \texttt{lmlasso} object comprising information about the variance components parameters and the number of nonzero fixed-effects coefficients.

Usage

\begin{verbatim}
## S3 method for class 'lmlasso'
print(x, ...)
\end{verbatim}

Arguments

\begin{description}
\item[x] \texttt{a \texttt{lmlasso} object}
\item[...] \texttt{not used}
\end{description}

See Also

\texttt{summary.lmlasso}

Examples

\begin{verbatim}
data(classroomstudy)
fit <-
lmlasso(x=classroomstudyX,y=classroomstudyY,z=classroomstudyZ,
grp=classroomstudy$grp,lambda=15,pdMat="pdIdent")
print(fit)
\end{verbatim}

\texttt{summary.lmlasso}

Summarize an \texttt{lmlasso} object

Description

Providing an elaborate summary of a \texttt{lmlasso} object.

Usage

\begin{verbatim}
## S3 method for class 'lmlasso'
summary(object, ...)
\end{verbatim}

Arguments

\begin{description}
\item[object] \texttt{a \texttt{lmlasso} object}
\item[...] \texttt{not used.}
\end{description}
Details

This function shows a detailed summary of a `lmmlasso` object. In the fixed-effects part, (n) right from a fixed-effects coefficient means that this coefficient was not subject to penalization.

Examples

```r
data(classroomStudy)
fit <- lmmlasso(x=classroomStudy$X, y=classroomStudy$y, z=classroomStudy$Z, grp=classroomStudy$grp, lambda=15, pdMat="pdIdent")
summary(fit)
```
Index

* Topic datasets
  classroomStudy, 2
* Topic hplot
  plot.lmmlasso, 8
* Topic methods
  summary.lmmlasso, 9
* Topic misc
  lmmlassoControl, 7
* Topic models
  lmmlasso, 3
* Topic package
  lmmlasso-package, 2
* Topic regression
  lmmlasso, 3
  classroomStudy, 2
  lmmlasso, 3
  lmmlasso-package, 2
  lmmlassoControl, 7
  plot.lmmlasso, 8
  print.lmmlasso, 9
  summary.lmmlasso, 9