

# Package ‘squeezy’

May 13, 2022

**Type** Package

**Title** Group-Adaptive Elastic Net Penalised Generalised Linear Models

**Version** 1.1-1

**Date** 2022-05-13

**Author** Mirrelijn M. van Nee [aut, cre],  
Tim van de Brug [aut],  
Mark A. van de Wiel [aut]

**Maintainer** Mirrelijn M. van Nee <m.vannee@amsterdamumc.nl>

**Depends** R (>= 3.5.0)

**Imports** glmnet, stats, Matrix, multiridge (>= 1.5), mvtnorm

**Suggests** ggplot2, ecpc

## Description

Fit linear and logistic regression models penalised with group-adaptive elastic net penalties. The group penalties correspond to groups of covariates defined by a co-data group set. The method accommodates inclusion of unpenalised covariates and overlapping groups. See Van Nee et al. (2021) <[arXiv:2101.03875](https://arxiv.org/abs/2101.03875)>.

**License** GPL (>= 3)

**URL** <https://arxiv.org/abs/2101.03875>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-05-13 14:20:02 UTC

## R topics documented:

squeezy-package	2
dminML.LA.ridgeGLM	2
mAIC.LA.ridgeGLM	4
minML.LA.ridgeGLM	5
normalityCheckQQ	6
squeezy	7

<b>Index</b>	<b>12</b>
--------------	-----------

---

squeezy-package

*Group-Adaptive Elastic Net Penalised Generalised Linear Models*

---

### Description

Fit linear and logistic regression models penalised with group-adaptive elastic net penalties. The group penalties correspond to groups of covariates defined by a co-data group set. The method accommodates inclusion of unpenalised covariates and overlapping groups. See Van Nee et al. (2021) <arXiv:2101.03875>.

### Details

See [squeezy](#) for example code.

### Author(s)

Mirrelijm M. van Nee [aut, cre], Tim van de Brug [aut], Mark A. van de Wiel [aut]

Maintainer: Mirrelijm M. van Nee <m.vanee@amsterdamumc.nl>

### References

Mirrelijm M. van Nee, Tim van de Brug, Mark A. van de Wiel, "Fast marginal likelihood estimation of penalties for group-adaptive elastic net.", 2021

---

dminML.LA.ridgeGLM

*Partial derivatives of -log(ML) of ridge penalised GLMs*

---

### Description

Returns the partial derivatives (w.r.t. 'loglambdas') of the minus log Laplace approximation (LA) of the marginal likelihood of ridge penalised generalised linear models. Note: currently only implemented for linear and logistic regression.

### Usage

```
dminML.LA.ridgeGLM(loglambdas, XXblocks, Y, sigmasq = 1,  
                   Xunpen = NULL, intrcpt = TRUE, model, minlam = 0,  
                   opt.sigma = FALSE)
```

**Arguments**

loglambdas	Logarithm of the ridge penalties as returned by ecpc or squeezey; Gx1 vector.
XXblocks	List of sample covariance matrices $X_g$ $\%*\%$ $t(X_g)$ for groups $g = 1, \dots, G$ .
Y	Response data; n-dimensional vector (n: number of samples) for linear and logistic outcomes.
sigmasq	(linear model only) Noise level ( $Y \sim N(X*\beta, sd=\sqrt{sigmasq})$ ).
Xunpen	Unpenalised variables; nxp_1-dimensional matrix for p_1 unpenalised variables.
intrcpt	Should an intercept be included? Set to TRUE by default.
model	Type of model for the response; linear or logistic.
minlam	Minimum value of lambda that is added to $\exp(\text{loglambdas})$ ; set to 0 as default.
opt.sigma	(linear model only) TRUE/FALSE if $\log(\text{sigmasq})$ is given as first argument of loglambdas for optimisation purposes

**Value**

Partial derivatives of the Laplace approximation of the minus log marginal likelihood to the model parameters 'loglambdas';

For opt.sigma=FALSE: Gx1-dimensional vector for the G log(group ridge penalties).

For opt.sigma=TRUE (linear model only): (G+1)x1-dimensional vector for the partial derivative to  $\log(\text{sigmasq})$  (first element) and for the G log(group ridge penalties).

**Examples**

```
#Simulate toy data
n<-100
p<-300
X <- matrix(rnorm(n*p),n,p)
Y <- rnorm(n)
groupset <- list(1:(p/2),(p/2+1):p)
sigmahat <- 2
alpha <- 0.5
tauMR <- c(0.01,0.005)

XXblocks <- lapply(groupset, function(x)X[,x]%*%t(X[,x]))

#compute partial derivatives of the minus log marginal likelihood to the penalties only
dminML.LA.ridgeGLM(loglambdas = log(sigmahat/tauMR),
  XXblocks, Y, sigmasq = sigmahat,
  model="linear",opt.sigma=FALSE)

#additionally, compute the partial derivative to the linear regression noise parameter sigma^2
dminML.LA.ridgeGLM(loglambdas = log(c(sigmahat,sigmahat/tauMR)),
  XXblocks, Y, sigmasq = sigmahat,
  model="linear",opt.sigma=TRUE)
```

mAIC.LA.ridgeGLM

*Marginal AIC of a multi-group, ridge penalised GLM***Description**

Compute the marginal AIC for the marginal likelihood (ML) of multi-group, ridge penalised generalised linear models. Note: currently only implemented for linear and logistic regression.

**Usage**

```
mAIC.LA.ridgeGLM(loglambdas, XXblocks, Y, sigmasq = 1,
                 Xunpen = NULL, intrcpt = TRUE, model, minlam = 0)
```

**Arguments**

loglambdas	Logarithm of the ridge penalties as returned by ecpc or squeezey; Gx1 vector.
XXblocks	List of sample covariance matrices $X_g \%*\% t(X_g)$ for groups $g = 1, \dots, G$ .
Y	Response data; n-dimensional vector (n: number of samples) for linear and logistic outcomes.
sigmasq	(linear model only) Noise level ( $Y \sim N(X*\beta, sd=\sqrt{sigmasq})$ ).
Xunpen	Unpenalised variables; nxp_1-dimensional matrix for p_1 unpenalised variables.
intrcpt	Should an intercept be included? Set to TRUE by default.
model	Type of model for the response; linear or logistic.
minlam	Minimum value of lambda that is added to $\exp(\text{loglambdas})$ ; set to 0 as default.

**Value**

mAIC	mAIC of the model
------	-------------------

**Examples**

```
#Simulate toy data
n<-100
p<-300
X <- matrix(rnorm(n*p),n,p)
Y <- rnorm(n)
groupset <- list(1:(p/2),(p/2+1):p)
sigmahat <- 2
alpha <- 0.5
tauMR <- c(0.01,0.005)

XXblocks <- lapply(groupset, function(x)X[,x]%%t(X[,x]))

#compute the mAIC of a co-data model with multiple groups
mAIC.LA.ridgeGLM(loglambdas=log(sigmahat/tauMR), XXblocks=XXblocks,
                 Y = Y, sigmasq = sigmahat, model="linear")
```

```
#compute the mAIC of a co-data agnostic model, i.e. only one group of covariates
mAIC.LA.ridgeGLM(loglambdas=log(sigmahat/median(tauMR)),
                 XXblocks=list(X%*%t(X)),
                 Y = Y, sigmasq = sigmahat, model="linear")
```

---

minML.LA.ridgeGLM      *-log(ML) of ridge penalised GLMs*

---

## Description

Returns the Laplace approximation (LA) of the minus log marginal likelihood of ridge penalised generalised linear models. Note: currently only implemented for linear and logistic regression.

## Usage

```
minML.LA.ridgeGLM(loglambdas, XXblocks, Y, sigmasq = 1,
                  Xunpen = NULL, intrcpt = TRUE, model, minlam = 0,
                  opt.sigma = FALSE)
```

## Arguments

loglambdas	Logarithm of the ridge penalties as returned by ecpc or squeezey; Gx1 vector.
XXblocks	List of sample covariance matrices $X_g \%*\% t(X_g)$ for groups $g = 1, \dots, G$ .
Y	Response data; n-dimensional vector (n: number of samples) for linear and logistic outcomes.
sigmasq	(linear model only) Noise level ( $Y \sim N(X * \beta, sd = \sqrt{\text{sigmasq}})$ ).
Xunpen	Unpenalised variables; nxp_1-dimensional matrix for p_1 unpenalised variables.
intrcpt	Should an intercept be included? Set to TRUE by default.
model	Type of model for the response; linear or logistic.
minlam	Minimum value of lambda that is added to $\exp(\text{loglambdas})$ ; set to 0 as default.
opt.sigma	(linear model only) TRUE/FALSE if $\log(\text{sigmasq})$ is given as first argument of loglambdas for optimisation purposes

## Value

Laplace approximation of the minus log marginal likelihood for the ridge penalised GLM with model parameters 'loglambdas' and 'sigmasq' (for linear regression).

**Examples**

```
#Simulate toy data
n<-100
p<-300
X <- matrix(rnorm(n*p),n,p)
Y <- rnorm(n)
groupset <- list(1:(p/2),(p/2+1):p)
sigmahat <- 2
alpha <- 0.5
tauMR <- c(0.01,0.005)

XXblocks <- lapply(groupset, function(x)X[,x]%*%t(X[,x]))

#compute minus log marginal likelihood
minML.LA.ridgeGLM(loglambdas = log(sigmahat/tauMR),
                  XXblocks, Y, sigmasq = sigmahat,
                  model="linear")
```

---

normalityCheckQQ	<i>Visual posterior check of multivariate normality of the linear predictors</i>
------------------	--

---

**Description**

Produce a qq-plot to visually check whether the assumption of multivariate normality of the linear predictors is valid for the data and model fit with 'squeezy'.

**Usage**

```
normalityCheckQQ(X,groupset,fit.squeezy,nSim=500)
```

**Arguments**

X	Observed data; (n $\times$ p)-dimensional matrix (p: number of covariates) with each row the observed high-dimensional feature vector of a sample.
groupset	Co-data group set; list with G groups. Each group is a vector containing the indices of the covariates in that group.
fit.squeezy	Model fit obtained by the function <a href="#">squeezy</a> .
nSim	Number of simulated vectors of linear predictors.

**Value**

The qqplot of the empirical versus theoretical quantiles is plotted. If 'ggplot2' is installed, the plot is returned as 'ggplot' object.

**Examples**

```
#Simulate toy data
n<-100
p<-300
X <- matrix(rnorm(n*p),n,p)
Y <- rnorm(n)
groupset <- list(1:(p/2),(p/2+1):p)
sigmahat <- 2
alpha <- 0.5
tauMR <- c(0.01,0.005)

#Fit group-regularised elastic net model with squeezy
fit.squeezy <- squeezy(Y,X,groupset,alpha=alpha,
                      lambdas=sigmahat/tauMR,sigmasq=sigmahat,
                      lambdaglobal=mean(sigmahat/tauMR))

#Check qq-plot
normalityCheckQQ(X,groupset,fit.squeezy)
```

squeezy

*Fit a group-adaptive elastic net penalised linear or logistic model***Description**

Estimate group-specific elastic net penalties and fit a linear or logistic regression model.

**Usage**

```
squeezy(Y, X, groupset, alpha = 1, model = NULL, X2 = NULL,
        Y2 = NULL, unpen = NULL, intrcpt = TRUE,
        method = c("ecpcEN", "MML", "MML.noDeriv", "CV"),
        fold = 10, compareMR = TRUE, selectAIC = FALSE, fit.ecpc = NULL,
        lambdas = NULL, lambdaglobal = NULL, lambdasinit = NULL,
        sigmasq = NULL, ecpcinit = TRUE, SANN = FALSE, minlam = 10^-3,
        standardise_Y = NULL, reCV = NULL, opt.sigma = NULL,
        resultsAICboth = FALSE, silent=FALSE)
```

**Arguments**

Y	Response data; n-dimensional vector (n: number of samples) for linear and logistic outcomes.
X	Observed data; (n x p)-dimensional matrix (p: number of covariates) with each row the observed high-dimensional feature vector of a sample.
groupset	Co-data group set; list with G groups. Each group is a vector containing the indices of the covariates in that group.
alpha	Elastic net penalty mixing parameter.
model	Type of model for the response; linear or logistic.

X2	(optional) Independent observed data for which response is predicted.
Y2	(optional) Independent response data to compare with predicted response.
unpen	Unpenalised covariates; vector with indices of covariates that should not be penalised.
intrcpt	Should an intercept be included? Included by default for linear and logistic, excluded for Cox for which the baseline hazard is estimated.
method	Which method should be used to estimate the group-specific penalties? Default MML.
fold	Number of folds used in inner cross-validation to estimate (initial) global ridge penalty lambda (if not given).
compareMR	TRUE/FALSE to fit the multi-ridge model and return results for comparison.
selectAIC	TRUE/FALSE to select the single-group model or multi-group model.
fit.ecpc	(optional) Model fit obtained by the function ecpc (from the ecpc R-package)
lambdas	(optional) Group-specific ridge penalty parameters. If given, these are transformed to elastic net penalties.
lambdaglobal	(optional) Global ridge penalty parameter used for initialising the optimisation.
lambdasinit	(optional) Group-specific ridge penalty parameters used for initialising the optimisation.
sigmasq	(linear model only) If given, noise level is fixed ( $Y \sim N(X * \beta, \text{sd} = \sqrt{\text{sigmasq}})$ ).
ecpcinit	TRUE/FALSE for using group-specific ridge penalties as given in 'fit.ecpc' for initialising the optimisation.
SANN	('method'=MML.noDeriv only) TRUE/FALSE to use simulated annealing in optimisation of the ridge penalties.
minlam	Minimal value of group-specific ridge penalty used in the optimisation.
standardise_Y	TRUE/FALSE should Y be standardised?
reCV	TRUE/FALSE should the elastic net penalties be recalibrated by cross-validation of a global rescaling penalty?
opt.sigma	(linear model only) TRUE/FALSE to optimise sigmasq jointly with the ridge penalties.
resultsAICboth	(selectAIC=TRUE only) TRUE/FALSE to return results of both the single-group and multi-group model.
silent	Should output messages be suppressed (default FALSE)?

### Value

betaApprox	Estimated regression coefficients of the group-adaptive elastic net model; p-dimensional vector.
a0Approx	Estimated intercept of the group-adaptive elastic net model; scalar.
lambdaApprox	Estimated group penalty parameters of the group-adaptive elastic net model; G-dimensional vector.
lambdapApprox	Estimated elastic net penalty parameter of the group-adaptive elastic net model for all covariates; p-dimensional vector.



<code>tauMR</code>	Estimated group variances of the multi-ridge model; G-dimensional vector.
<code>lambdaMR</code>	Estimated group penalties of the multi-ridge model; G-dimensional vector.
<code>lambdaglobal</code>	Estimated global ridge penalty; scalar. Note: only optimised if <code>selectAIC=TRUE</code> or <code>compareMR=TRUE</code> , else the returned crude estimate is sufficient for initialisation of <code>squeezy</code> .
<code>sigmahat</code>	(linear model) Estimated $\sigma^2$ ; scalar.
<code>MLinit</code>	Min log marginal likelihood value at initial group penalties; scalar.
<code>MLfinal</code>	Min log marginal likelihood value at estimated group penalties; scalar.
<code>alpha</code>	Value used for the elastic net mixing parameter <code>alpha</code> ; scalar.
<code>glmnet.fit</code>	Fit of the 'glmnet' function to obtain the regression coefficients.

If `'compareMR'=TRUE`, multi-ridge model is returned as well:

<code>betaMR</code>	Estimated regression coefficients of the multi-ridge model; p-dimensional vector.
<code>a0MR</code>	Estimated intercept of the multi-ridge model; scalar.

If independent test set 'X2' is given, predictions and MSE are returned:

<code>YpredApprox</code>	Predictions for the test set of the estimated group-adaptive elastic net model.
<code>MSEApprox</code>	Mean squared error on the test set of the estimated group-adaptive elastic net model.
<code>YpredMR</code>	Predictions for the test set of the estimated group-adaptive multi-ridge model.
<code>MSEMR</code>	Mean squared error on the test set of the estimated group-adaptive multi-ridge model.

If `'selectAIC'=TRUE`, the multi-group or single-group model with best AIC is selected. Results in `'betaApprox'`, `'a0Approx'`, `'lambdaApprox'` contain those results of the best model. Summary results of both models are included as well:

<code>AICmodels</code>	List with elements "multigroup" and "onegroup".- Each element is a list with results of the multi-group or single-group model, containing the group penalties ( <code>'lambdas'</code> ), $\sigma^2$ ( <code>'sigmahat'</code> , linear model only), and AIC ( <code>'AIC'</code> ). If besides <code>'selectAIC'=TRUE</code> , also <code>'resultsAICboth'=TRUE</code> , the fit of both the single-group model and multi-group model as obtained with <code>squeezy</code> are returned ( <code>'fit'</code> ).
<code>modelbestAIC</code>	Either "onegroup" or "multigroup" for the selected model.

### Author(s)

Mirrelijm M. van Nee, Tim van de Brug, Mark A. van de Wiel

### References

Mirrelijm M. van Nee, Tim van de Brug, Mark A. van de Wiel, "Fast marginal likelihood estimation of penalties for group-adaptive elastic net", arXiv preprint, arXiv:2101.03875 (2021).

## Examples

```
#####
# Simulate toy data #
#####
p<-100 #number of covariates
n<-50 #sample size training data set
n2<-100 #sample size test data set
G<- 5 #number of groups

taugrp <- rep(c(0.05,0.1,0.2,0.5,1),each=p/G) #ridge prior variance
groupIndex <- rep(1:G,each=p/G) #groups for co-data
groupset <- lapply(1:G,function(x){which(groupIndex==x)}) #group set with each element one group
sigmasq <- 2 #linear regression noise
lambda1 <- sqrt(taugrp/2) #corresponding lasso penalty
#A Laplace(0,b) variate can also be generated as the difference of two i.i.d.
#Exponential(1/b) random variables
betas <- rexp(p, 1/lambda1) - rexp(p, 1/lambda1) #regression coefficients
X <- matrix(rnorm(n*p),n,p) #simulate training data
Y <- rnorm(n,X%*betas,sd=sqrt(sigmasq))
X2 <- matrix(rnorm(n*p),n,p) #simulate test data
Y2 <- rnorm(n,X2%*betas,sd=sqrt(sigmasq))

#####
# Fit squeezy #
#####
#may be fit directly..
res.squeezy <- squeezy(Y,X,groupset=groupset,Y2=Y2,X2=X2,
                      model="linear",alpha=0.5)

#..or with ecpc-fit as initialisation
if(requireNamespace("ecpc")){
  res.ecpc <- ecpc::ecpc(Y,X, #observed data and response to train model
                        groupsets=list(groupset), #informative co-data group set
                        Y2=Y2,X2=X2, #test data
                        model="linear",
                        hypershrinkage="none",postselection = FALSE)
  res.squeezy <- squeezy(Y,X, #observed data and response to train model
                        groupset=groupset, #informative co-data group set
                        Y2=Y2,X2=X2, #test data
                        fit.ecpc = res.ecpc, #ecpc-fit for initial values
                        model="linear", #type of model for the response
                        alpha=0.5) #elastic net mixing parameter
}

summary(res.squeezy$betaApprox) #estimated elastic net regression coefficients
summary(res.squeezy$betaMR) #estimated multi-ridge regression coefficients
res.squeezy$lambdaApprox #estimated group elastic net penalties
res.squeezy$tauMR #multi-ridge group variances
```

```
res.squeezy$MSEApprox #MSE group-elastic net model
res.squeezy$MSEMR #MSE group-ridge model

#once fit, quickly find model fit for different values of alpha:
res.squeezy2 <- squeezy(Y,X, #observed data and response to train model
  groupset=groupset, #informative co-data groupset
  Y2=Y2,X2=X2, #test data
  lambdas = res.squeezy$lambdaMR, #fix lambdas at multi-ridge estimate
  model="linear", #type of model for the response
  alpha=0.9) #elastic net mixing parameter

#Select single-group model or multi-group model based on best mAIC
res.squeezy <- squeezy(Y,X, #observed data and response to train model
  groupset=groupset, #informative co-data group set
  Y2=Y2,X2=X2, #test data
  fit.ecpc = res.ecpc, #ecpc-fit for initial values
  model="linear", #type of model for the response
  alpha=0.5, #elastic net mixing parameter
  selectAIC = TRUE,resultsAICboth = TRUE)

res.squeezy$modelbestAIC #selected model
res.squeezy$AICmodels$multigroup$fit$MSEApprox #MSE on test set of multi-group model
res.squeezy$AICmodels$onegroup$fit$MSEApprox #MSE on test set of single-group model
```

# Index

`dminML.LA.ridgeGLM`, [2](#)

`mAIC.LA.ridgeGLM`, [4](#)

`minML.LA.ridgeGLM`, [5](#)

`normalityCheckQQ`, [6](#)

`squeezy`, [2](#), [6](#), [7](#), [9](#)

`squeezy-package`, [2](#)