Package 'smoothedLasso'

May 9, 2026

Type Package

Title A Framework to Smooth L1 Penalized Regression Operators using Nesterov Smoothing

Version 1.6

Date 2021-03-18

Author Georg Hahn [aut,cre], Sharon M. Lutz [ctb], Nilanjana Laha [ctb], Christoph Lange [ctb]

Maintainer Georg Hahn <ghahn@hsph.harvard.edu>

Description

We provide full functionality to smooth L1 penalized regression operators and to compute regression estimates thereof. For this, the objective function of a user-specified regression operator is first smoothed using Nesterov smoothing (see Y. Nesterov (2005) <doi:10.1007/s10107-004-0552-5>), resulting in a modified objective function with explicit gradients everywhere. The smoothed objective function and its gradient are minimized via BFGS, and the obtained minimizer is returned. Using Nesterov smoothing, the smoothed objective function can be made arbitrarily close to the original (unsmoothed) one. In particular, the Nesterov approach has the advantage that it comes with explicit accuracy bounds, both on the L1/L2 difference of the unsmoothed to the smoothed objective functions as well as on their respective minimizers (see G. Hahn, S.M. Lutz, N. Laha, C. Lange (2020) <doi:10.1101/2020.09.17.301788>). A progressive smoothing approach is provided which iteratively smoothes the objective function, resulting in more stable regression estimates. A function to perform cross validation for selection of the regularization parameter is provided.

License GPL (>= 2)

Imports Rdpack, Matrix

RdMacros Rdpack

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-03-21 07:20:02 UTC

Contents

`crossvalidation` *Perform cross validation to select the regularization parameter.*

Description

Perform cross validation to select the regularization parameter.

Usage

```
crossvalidation(auxfun, X, y, param, K = 10)
```

Arguments

`auxfun`	A complete fitting function which takes as arguments a data matrix X, a response vector y, and some parameter p (to be tuned), and returns the estimator ($betavector$) (minimizer) of the regression operator under investigation.
`X`	The design matrix.
`y`	The response vector.
`param`	A vector of regularization parameters which are to be evaluated via cross validation.
`K`	The number of folds for cross validation (should divide the number of rows of X). The default is 10.

Value

A vector of average errors over all folds. The entries in the returned vector correspond to the entries in the vector $param$ in the same order.

References

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Tibshirani, R. (2013). Model selection and validation 1: Cross-validation. https://www.stat.cmu.edu/~ryantibs/datamining/lec
val1.pdf

Examples

```
library(smoothedLasso)
n <- 1000
p <- 100
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
auxfun <- function(X,y,lambda) {
temp <- standardLasso(X,y,lambda)
obj <- function(z) objFunction(z,temp$u,temp$v,temp$w)
objgrad <- function(z) objFunctionGradient(z,temp$w,temp$du,temp$dv,temp$dw)
return(minimizeFunction(p,obj,objgrad))
}
lambdaVector <- seq(0,1,by=0.1)
print(crossvalidation(auxfun,X,y,lambdaVector,10))
```

---

elasticNet *Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the elastic net.*

Description

Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the elastic net.

Usage

```
elasticNet(X, y, alpha)
```

Arguments

X	The design matrix.
y	The response vector.
alpha	The regularization parameter of the elastic net.

Value

A list with six functions, precisely the objective u, penalty v, and dependence structure w, as well as their derivatives du, dv, and dw.

References

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. J Roy Stat Soc B Met, 67(2):301-320.

Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Tay, K., Simon, N., and Qian, J. (2020). glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models. R-package version 4.0.

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
alpha <- 0.5
temp <- elasticNet(X,y,alpha)
```

`fusedLasso` *Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the fused Lasso.*

Description

Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the fused Lasso.

Usage

```
fusedLasso(X, y, E, lambda, gamma)
```

Arguments

`X`	The design matrix.
`y`	The response vector.
`E`	The adjacency matrix which encodes with a one in position (i, j) the presence of an edge between variables i and j. Note that only the upper triangle of E is read.
`lambda`	The first regularization parameter of the fused Lasso.
`gamma`	The second regularization parameter of the fused Lasso.

Value

A list with six functions, precisely the objective u, penalty v, and dependence structure w, as well as their derivatives du, dv, and dw.

References

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and Smoothness via the Fused Lasso. J Roy Stat Soc B Met, 67(1):91-108.

Arnold, T.B. and Tibshirani, R.J. (2020). genlasso: Path Algorithm for Generalized Lasso Problems. R package version 1.5.

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
E <- matrix(sample(c(TRUE,FALSE),p*p,replace=TRUE),p)
lambda <- 1
gamma <- 0.5
temp <- fusedLasso(X,y,E,lambda,gamma)
```

`graphicalLasso` *Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the graphical Lasso.*

Description

Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the graphical Lasso.

Usage

```
graphicalLasso(S, lambda)
```

Arguments

`S`	The sample covariance matrix.
`lambda`	The regularization parameter of the graphical Lasso.

Value

A list with three functions, precisely the objective u, penalty v, and dependence structure w. Not all derivatives are available in closed form, and thus computing the numerical derivative of the entire objective function is recommended.

References

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9(3):432-441.

Friedman, J., Hastie, T., and Tibshirani, R. (2019). glasso: Graphical Lasso: Estimation of Gaussian Graphical Models. R package version 1.11.

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
p <- 30
S <- matrix(rWishart(1,p,diag(p)),p)
lambda <- 1
temp <- graphicalLasso(S,lambda)
```

`minimizeFunction` *Minimize the objective function of an unsmoothed or smoothed regression operator with respect to* $betavector$ *using BFGS.*

Description

Minimize the objective function of an unsmoothed or smoothed regression operator with respect to $betavector$ using BFGS.

Usage

```
minimizeFunction(p, obj, objgrad)
```

Arguments

`p`	The dimension of the unknown parameters (regression coefficients).
`obj`	The objective function of the regression operator as a function of $betavector$.
`objgrad`	The gradient function of the regression operator as a function of $betavector$.

Value

The estimator $betavector$ (minimizer) of the regression operator.

References

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
lambda <- 1
temp <- standardLasso(X,y,lambda)
obj <- function(z) objFunctionSmooth(z,temp$u,temp$v,temp$w,mu=0.1)
objgrad <- function(z) objFunctionSmoothGradient(z,temp$w,temp$du,temp$dv,temp$dw,mu=0.1)
print(minimizeFunction(p,obj,objgrad))
```

`minimizeSmoothedSequence` *Minimize the objective function of a smoothed regression operator with respect to* $betavector$ *using the progressive smoothing algorithm.*

Description

Minimize the objective function of a smoothed regression operator with respect to $betavector$ using the progressive smoothing algorithm.

Usage

```
minimizeSmoothedSequence(p, obj, objgrad, muSeq = 2^seq(3, -6))
```

Arguments

`p`	The dimension of the unknown parameters (regression coefficients).
`obj`	The objective function of the regression operator. Note that in the case of the progressive smoothing algorithm, the objective function must be a function of both $betavector$ and mu.
`objgrad`	The gradient function of the regression operator. Note that in the case of the progressive smoothing algorithm, the gradient must be a function of both $betavector$ and mu.
`muSeq`	The sequence of Nesterov smoothing parameters. The default is 2^{-n} for $n \in \{-3, \ldots, 6\}$.

Value

The estimator $betavector$ (minimizer) of the regression operator.

References

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
lambda <- 1
temp <- standardLasso(X,y,lambda)
obj <- function(z,m) objFunctionSmooth(z,temp$u,temp$v,temp$w,mu=m)
objgrad <- function(z,m) objFunctionSmoothGradient(z,temp$w,temp$du,temp$dv,temp$dw,mu=m)
print(minimizeSmoothedSequence(p,obj,objgrad))
```

objFunction *Auxiliary function to define the objective function of an L1 penalized regression operator.*

Description

Auxiliary function to define the objective function of an L1 penalized regression operator.

Usage

```
objFunction(betavector, u, v, w)
```

Arguments

betavector	The vector of regression coefficients.
u	The function encoding the objective of the regression operator.
v	The function encoding the penalty of the regression operator.
w	The function encoding the dependence structure among the regression coefficients.

Value

The value of the L1 penalized regression operator for the input $betavector$.

References

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
lambda <- 1
temp <- standardLasso(X,y,lambda)
print(objFunction(betavector,temp$u,temp$v,temp$w))
```

`objFunctionGradient` *Auxiliary function which computes the (non-smooth) gradient of an L1 penalized regression operator.*

Description

Auxiliary function which computes the (non-smooth) gradient of an L1 penalized regression operator.

Usage

```
objFunctionGradient(betavector, w, du, dv, dw)
```

Arguments

`betavector`	The vector of regression coefficients.
`w`	The function encoding the dependence structure among the regression coefficients.
`du`	The derivative (gradient) of the objective of the regression operator.
`dv`	The derivative (gradient) of the penalty of the regression operator.
`dw`	The derivative (Jacobian matrix) of the function encoding the dependence structure among the regression coefficients.

Value

The value of the gradient for the input $betavector$.

References

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
lambda <- 1
temp <- standardLasso(X,y,lambda)
print(objFunctionGradient(betavector,temp$w,temp$du,temp$dv,temp$dw))
```

`objFunctionSmooth` *Auxiliary function to define the objective function of the smoothed L1 penalized regression operator.*

Description

Auxiliary function to define the objective function of the smoothed L1 penalized regression operator.

Usage

```
objFunctionSmooth(betavector, u, v, w, mu, entropy = TRUE)
```

Arguments

`betavector`	The vector of regression coefficients.
`u`	The function encoding the objective of the regression operator.
`v`	The function encoding the penalty of the regression operator.
`w`	The function encoding the dependence structure among the regression coefficients.
`mu`	The Nesterov smoothing parameter.
`entropy`	A boolean switch to select the entropy prox function (default) or the squared error prox function.

Value

The value of the smoothed regression operator for the input $betavector$.

References

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
lambda <- 1
temp <- standardLasso(X,y,lambda)
print(objFunctionSmooth(betavector,temp$u,temp$v,temp$w,mu=0.1))
```

objFunctionSmoothGradient *Auxiliary function which computes the gradient of the smoothed L1 penalized regression operator.*

Description

Auxiliary function which computes the gradient of the smoothed L1 penalized regression operator.

Usage

```
objFunctionSmoothGradient(betavector, w, du, dv, dw, mu, entropy = TRUE)
```

Arguments

betavector	The vector of regression coefficients.
w	The function encoding the dependence structure among the regression coefficients.
du	The derivative (gradient) of the objective of the regression operator.
dv	The derivative (gradient) of the penalty of the regression operator.
dw	The derivative (Jacobian matrix) of the function encoding the dependence structure among the regression coefficients.
mu	The Nesterov smoothing parameter.
entropy	A boolean switch to select the entropy prox function (default) or the squared error prox function.

Value

The value of the gradient for the input $betavector$.

References

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
lambda <- 1
temp <- standardLasso(X,y,lambda)
print(objFunctionSmoothGradient(betavector,temp$w,temp$du,temp$dv,temp$dw,mu=0.1))
```

`prsLasso` *Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the Lasso for polygenic risk scores (prs).*

Description

Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the Lasso for polygenic risk scores (prs).

Usage

```
prsLasso(X, y, s, lambda)
```

Arguments

`X`	The design matrix.
`y`	The response vector.
`s`	The shrinkage parameter used to regularize the design matrix.
`lambda`	The regularization parameter of the prs Lasso.

Value

A list with six functions, precisely the objective u, penalty v, and dependence structure w, as well as their derivatives du, dv, and dw.

References

Mak, T.S., Porsch, R.M., Choi, S.W., Zhou, X., and Sham, P.C. (2017). Polygenic scores via penalized regression on summary statistics. Genet Epidemiol, 41(6):469-480.

Mak, T.S. and Porsch, R.M. (2020). lassosum: LASSO with summary statistics and a reference panel. R package version 0.4.5.

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
s <- 0.5
lambda <- 1
temp <- prsLasso(X,y,s,lambda)
```

`standardLasso` *Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the Lasso.*

Description

Auxiliary function which returns the objective, penalty, and dependence structure among regression coefficients of the Lasso.

Usage

```
standardLasso(X, y, lambda)
```

Arguments

`X`	The design matrix.
`y`	The response vector.
`lambda`	The Lasso regularization parameter.

Value

A list with six functions, precisely the objective u, penalty v, and dependence structure w, as well as their derivatives du, dv, and dw.

References

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. J Roy Stat Soc B Met, 58(1):267-288.

Hahn, G., Lutz, S., Laha, N., and Lange, C. (2020). A framework to efficiently smooth L1 penalties for linear regression. bioRxiv:2020.09.17.301788.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
betavector <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% betavector
lambda <- 1
temp <- standardLasso(X,y,lambda)
```

Index