Package 'smoothedLasso'

June 14, 2020

Type Package

Title Smoothed LASSO Regression via Nesterov Smoothing

Version 1.3

Date 2020-06-11

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 compute smoothed LASSO regression estimates. For this, the LASSO objective function is first smoothed using Nesterov smoothing (see Y. Nesterov (2005) <doi:10.1007/s10107-004-0552-5>), resulting in a modified LASSO objective function with explicit gradients everywhere. The smoothed objective function and its gradient are used to minimize it via BFGS, and the obtained minimizer is returned. Using Nesterov smoothing, the smoothed LASSO 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 LASSO objective function as well as on their respective minimizers. A progressive smoothing approach is provided which iteratively smoothes the LASSO, resulting in more stable regression estimates.

License GPL (>= 2)

Imports Rdpack, Matrix

RdMacros Rdpack

RoxygenNote 7.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-14 16:30:02 UTC

R topics documented:

`objFunction` *Auxiliary function defining the LASSO objective function.*

Description

Auxiliary function defining the LASSO objective function.

Usage

```
objFunction(beta, X, y, lambda)
```

Arguments

`beta`	The p-vector of coefficients.
`X`	The data matrix of dimensions $n \times p$.
`y`	The n-vector of responses.
`lambda`	The LASSO regularization parameter.

Value

The value of the LASSO objective function.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
beta <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% beta
lambda <- 1
print(objFunction(beta,X,y,lambda))
```

objFunctionGradient — *Auxiliary function which computes the (non-smooth) gradient of the LASSO objective function with respect to β.*

Description

Auxiliary function which computes the (non-smooth) gradient of the LASSO objective function with respect to β.

Usage

```
objFunctionGradient(beta, X, y, lambda)
```

Arguments

beta	The p-vector of coefficients.
X	The data matrix of dimensions $n \times p$.
y	The n-vector of responses.
lambda	The LASSO regularization parameter.

Value

The value of the gradient of the LASSO objective function at β.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
beta <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% beta
lambda <- 1
print(objFunctionGradient(beta,X,y,lambda))
```

objFunctionSmooth — *Auxiliary function defining the smoothed LASSO objective function.*

Description

Auxiliary function defining the smoothed LASSO objective function.

Usage

```
objFunctionSmooth(beta, X, y, lambda, mu, entropy = T)
```

Arguments

beta	The p-vector of coefficients.
X	The data matrix of dimensions $n \times p$.
y	The n-vector of responses.
lambda	The LASSO regularization parameter.
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 LASSO objective function.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
beta <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% beta
lambda <- 1
print(objFunctionSmooth(beta,X,y,lambda,mu=0.1))
```

---

objFunctionSmoothGradient

Auxiliary function which computes the gradient of the smoothed LASSO objective function with respect to β.

Description

Auxiliary function which computes the gradient of the smoothed LASSO objective function with respect to β.

Usage

```
objFunctionSmoothGradient(beta, X, y, lambda, mu, entropy = T)
```

Arguments

beta	The p-vector of coefficients.
X	The data matrix of dimensions $n \times p$.
y	The n-vector of responses.
lambda	The LASSO regularization parameter.
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 of the LASSO objective function at β.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
beta <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% beta
lambda <- 1
print(objFunctionSmoothGradient(beta,X,y,lambda,mu=0.1))
```

---

solveSmoothedLASSO *Minimize the smoothed LASSO objective function with respect to β using BFGS.*

Description

Minimize the smoothed LASSO objective function with respect to β using BFGS.

Usage

```
solveSmoothedLASSO(X, y, lambda, mu, entropy = T)
```

Arguments

X	The data matrix of dimensions $n \times p$.
y	The n-vector of responses.
lambda	The LASSO regularization parameter.
mu	The Nesterov smoothing parameter.
entropy	A boolean switch to select the entropy prox function (default) or the squared error prox function.

Value

The LASSO estimator β.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
beta <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% beta
lambda <- 1
print(solveSmoothedLASSO(X,y,lambda,mu=0.1))
```

`solveSmoothedLASSOSequence`

Minimize the smoothed LASSO objective function with respect to β using the progressive smoothing algorithm.

Description

Minimize the smoothed LASSO objective function with respect to β using the progressive smoothing algorithm.

Usage

```
solveSmoothedLASSOSequence(X, y, lambda, muSeq = 2^seq(3, -6), entropy = T)
```

Arguments

`X`	The data matrix of dimensions $n \times p$.
`y`	The n-vector of responses.
`lambda`	The LASSO regularization parameter.
`muSeq`	The sequence of Nesterov smoothing parameters. The default is 2^{-n} for $n \in \{0, \ldots, 5\}$.
`entropy`	A boolean switch to select the entropy prox function (default) or the squared error prox function.

Value

The LASSO estimator β.

Examples

```
library(smoothedLasso)
require(Matrix)
n <- 100
p <- 500
beta <- runif(p)
X <- Matrix(sample(0:1,size=n*p,replace=TRUE),nrow=n,ncol=p,sparse=TRUE)
```

```
y <- X %*% beta
lambda <- 1
print(solveSmoothedLASSOSequence(X,y,lambda))
```

`solveUnsmoothedLASSO` *Minimize the unsmoothed LASSO objective function with respect to β. Three options are available: BFGS with analytical gradient ($method = 0$), BFGS with numerical gradient ($method = 1$), and simulated annealing which is gradient free ($method = 2$). The default is $method = 0$.*

Description

Minimize the unsmoothed LASSO objective function with respect to β. Three options are available: BFGS with analytical gradient ($method = 0$), BFGS with numerical gradient ($method = 1$), and simulated annealing which is gradient free ($method = 2$). The default is $method = 0$.

Usage

```
solveUnsmoothedLASSO(X, y, lambda, method = 0)
```

Arguments

`X`	The data matrix of dimensions $n \times p$.
`y`	The n-vector of responses.
`lambda`	The LASSO regularization parameter.
`method`	The method used for minimization: BFGS with analytical gradient ($method = 0$), BFGS with numerical gradient ($method = 1$), and simulated annealing which is gradient free ($method = 2$). The default is $method = 0$.

Value

The LASSO estimator β.

Examples

```
library(smoothedLasso)
n <- 100
p <- 500
beta <- runif(p)
X <- matrix(runif(n*p),nrow=n,ncol=p)
y <- X %*% beta
lambda <- 1
print(solveUnsmoothedLASSO(X,y,lambda))
```

Index