

# Package: lassoshooting (via r-universe)

October 14, 2024

**Version** 0.1.5-1.1

**Date** 2012-04-26

**Title** L1 Regularized Regression (Lasso) Solver using the Cyclic Coordinate Descent Algorithm aka Lasso Shooting

**Author** Tobias Abenius <Tobias.Abenius@Chalmers.se>

**Maintainer** Tobias Abenius <Tobias.Abenius@Chalmers.se>

**Depends** R (>= 2.12.0)

**Description** L1 regularized regression (Lasso) solver using the Cyclic Coordinate Descent algorithm aka Lasso Shooting is fast. This implementation can choose which coefficients to penalize. It support coefficient-specific penalties and it can take  $X'X$  and  $X'y$  instead of  $X$  and  $y$ .

**License** LGPL-3

**Repository** CRAN

**Date/Publication** 2022-04-04 09:25:56 UTC

**NeedsCompilation** yes

## Contents

lassoshooting . . . . .	1
softthresh . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

lassoshooting	<i>Lasso Shooting</i>
---------------	-----------------------

---

## Description

Efficient estimates of sparse regression coefficients with a lasso (L1) penalty

**Usage**

```
lassoshooting(X=NULL, y=NULL, lambda, XtX=NULL, Xty=NULL, thr=1.0e-6,
             maxit=1e4, nopenalize=NULL, penaltyweight=NULL, trace=0, ...)
```

**Arguments**

<code>X</code>	Design matrix: N by p matrix of p explanatory variables
<code>y</code>	vector of 1 response variable for N observations
<code>XtX</code>	$X'X$ , could be given together with $X'y$ instead of $X$ and $y$
<code>Xty</code>	$X'y$ , could be given together with $X'X$ instead of $X$ and $y$
<code>lambda</code>	(Non-negative) regularization parameter for lasso. $\lambda=0$ means no regularization.
<code>thr</code>	Threshold for convergence. Default value is $1e-4$ . Iterations stop when max absolute parameter change is less than <code>thr</code>
<code>maxit</code>	Maximum number of iterations of outer loop. Default 10,000
<code>nopenalize</code>	List of coefficients not to penalize starting at 0
<code>penaltyweight</code>	p weights, one per variable, will be multiplied by overall lambda penalty
<code>trace</code>	Level of detail for printing out information as iterations proceed. Default 0 – no information
<code>...</code>	Reserved for experimental options

**Details**

Estimates a sparse regression coefficient vector using a lasso (L1) penalty using the approach of cyclic coordinate descent. See references for details.

The solver does NOT include an intercept, add a column of ones to  $x$  if your data is not centered.

**Value**

A list with components

<code>coefficients</code>	Estimated regression coefficient vector
<code>iterations</code>	Number of iterations of outer loop used by algorithm
<code>delta</code>	Change in parameter value at convergence
<code>infnorm</code>	$\ X'y\ _\infty$

**Author(s)**

Tobias Abenius

## References

- Rebecka Jörnsten, Tobias Abenius, Teresia Kling, Linnéa Schmidt, Erik Johansson, Torbjörn Nordling, Bodil Nordlander, Chris Sander, Peter Gennemark, Keiko Funa, Björn Nilsson, Linda Lindahl, Sven Nelander. (2011) Network modeling of the transcriptional effects of copy number aberrations in glioblastoma. *Molecular Systems Biology* 7 (to appear)
- Friedman J, Hastie T, et al. (2007) Pathwise coordinate optimization. *Ann Appl Stat* 1: 302–332
- Fu WJ (1998) Penalized regressions: the bridge versus the lasso. *J Comput Graph Statist* 7: 397–416

## Examples

```
## Not run:
set.seed(42)

b <- seq(3,3,length=10)
n<-100;
p<-10;
X <- matrix(rnorm(n*p),n,p)
noise <- as.matrix(rnorm(n,sd=0.1))
y <- X

require(lassoshooting)
# FIXME: write proper example using R built in dataset
#add intercept column to the design matrix
Xdesign <- cbind(1,X)
lambda <- 20
#don't penalize the intercept
bhat <- lassoshooting(X=Xdesign,y=y,lambda=lambda,nopenalize=0)

#above equals below
bhat1 <- lassoshooting(X=Xdesign,y=y,lambda=2*lambda,penaltyweight=c(0,seq(0.5,0.5,length=p-1)))

T1 <- all(abs(bhat1-bhat) < 1e-20)

c <- 10
bhat2 <- lassoshooting(X=Xdesign,y=y, lambda=lambda, penaltyweight=c(0,1,1,1,1,1,c,c,c,c))

T2 <- all(bhat2[2:6] > bhat2[7:11])
T1 && T2

## End(Not run)
```

---

 softthresh

*Soft Threshold*


---

## Description

Soft threshold

**Usage**

```
softthresh(x, t)
```

**Arguments**

x	value
t	threshold

**Details**

Pass x through a soft threshold with parameter t.

**Value**

A numeric scalar

**Examples**

```
## Not run:  
t <- 0.4  
plot(sapply(seq(-2,2,by=0.1),function (x) softthresh(x, t)),type='l')  
  
## End(Not run)
```

# Index

- \* **graphs**
    - lassoshooting, 1
  - \* **models**
    - lassoshooting, 1
  - \* **multivariate**
    - lassoshooting, 1
  - \* **noise**
    - softthresh, 3
  - \* **selection**
    - lassoshooting, 1
  - \* **threshold**
    - softthresh, 3
- lassoshooting, 1
- softthresh, 3