

# Package ‘Lavash’

January 20, 2025

**Type** Package

**Title** Lava Estimation for the Sum of Sparse and Dense Signals

**Version** 1.0

**Date** 2018-02-09

**Author** Victor Chernozhukov [aut, cre], Christian Hansen [aut, cre], Yuan Liao [aut, cre], Jaeheon Jung [ctb, cre]

**Maintainer** Jaeheon Jung <jaeheonj81@gmail.com>

## Description

The lava estimation is a new technique to recover signals that is the sum of a sparse and dense signals. The post-lava method corrects the shrinkage bias of lava. For more information on the lava estimation, see Chernozhukov, Hansen, and Liao (2017) <[doi:10.1214/16-AOS1434](https://doi.org/10.1214/16-AOS1434)>.

**Depends** MASS, glmnet, pracma

**License** GPL-2

**Repository** CRAN

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Date/Publication** 2018-02-12 09:25:12 UTC

## Contents

Lavash . . . . .	1
<b>Index</b>	<b>4</b>

---

Lavash

*Lava Estimation for the Sum of Sparse and Dense Signals*

---

## Description

The lava estimation of linear regression models, which are suitable for estimating coefficients that can be represented by the sum of sparse and dense signals. The post-lava method corrects the shrinkage bias of lava. The model is  $Y=X*b+error$ , where  $b$  can be decomposed into  $b=dense+sparse$ . The method of lava solves the following problem:  $\min_{[b_2,b_1]} 1/n*|Y-X*(b_1+b_2)|_2^2+lambd_2*|b_2|_2^2+lambd_1*|b_1|_1$ . The estimator is  $b_2+b_1$ . Both tuning parameters are chosen using the K-fold cross-validation.

## Usage

```
Lavash(X, Y, K, Lambda1, Lambda2, method = c("profile", "iteration"), Maxiter = 50)
```

## Arguments

X	n by p data matrix, where n and p respectively denote the sample size and the number of regressors.
Y	n by 1 matrix of outcome.
K	the K fold cross validation.
Lambda1	a vector of candidate values to be evaluated for lambda1, in the cross validation.
Lambda2	a vector of candidate values to be evaluated for lambda2, in the cross validation.
method	choose between 'profile' and 'iteration'. 'profile' computes using the profiled lasso method. 'iteration' computes using iterating lasso and ridge.
Maxiter	the maximum number of iterations. the default value is 50. Only used when 'iteration' is selected.

## Details

We recommend using a relatively long vector of Lambda1 (e.g., 50 or 100 values), but a short vector of Lambda2 (e.g., within 10). Higher dimensions of Lambda2 substantially increase the computational time because a 'for' loop is called for lambda2. Two algorithms are used: (i) the profiled lasso method, building on the singular value decomposition of the design matrix, and runs fast. (whose computational cost is nearly the same as that of lasso) (ii) the iteration between lasso and ridge.

## Value

lava_dense	parameter estimate of the dense component using lava
lava_sparse	parameter estimate of the sparse component using lava
lava_estimate	equals lava_dense+lava_sparse: parameter estimate using lava
postlava_dense	parameter estimate of the dense component using post-lava
postlava_sparse	parameter estimate of the sparse component using post-lava
post_lava	equals postlava_dense+postlava_sparse: parameter estimate using post-lava
LAMBDA	[lambda1lava,lambda2lava, lambda1post, lambda2post]: These are the CV-chosen lambda1 and lambda2 tunings for lava and post-lava.

**Author(s)**

Victor Chernozhukov, Christian Hansen, Yuan Liao, Jaeheon Jung

**References**

Chernozhukov, V., Hansen, C., and Liao, Y. (2017) "A lava attack on the recovery of sums of dense and sparse signals", *Annals of Statistics*, 45, 39-76

**Examples**

```
n <- 10; p <- 5
b <- matrix(0,p,1); b[1,1] <- 3; b[2:p,1] <- 0.1
X <- randn(n,p)
Y <- X*%b+randn(n,1)
K<-5
iter<-50
Lambda2<-c(0.01, 0.07, 0.2, 0.7, 3,10,60,1000,2000)
Lambda1<-seq(0.01,6,6/50)

result<-Lavash(X,Y,K,Lambda1,Lambda2,method="profile", Maxiter = iter)

result$lava_dense
result$lava_sparse
result$lava_estimate
result$postlava_dense
result$postlava_sparse
result$post_lava
result$LAMBDA
```

# Index

\* **lava**  
    Lavash, [1](#)

Lavash, [1](#)