

# Package: DWLasso (via r-universe)

September 5, 2024

**Type** Package

**Title** Degree Weighted Lasso

**Version** 1.1

**Author** Nurgazy Sulaimanov, Sunil Kumar and Heinz Koepl

**Maintainer** Nurgazy Sulaimanov <nurgazy.sulaimanov@bcs.tu-darmstadt.de>

**Description** Infers networks with hubs using degree weighted Lasso method.

**License** GPL-2

**Encoding** UTF-8

**Imports** Matrix, glmnet, hglasso

**LazyData** true

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-10-06 07:59:10 UTC

## Contents

combined . . . . .	2
degreeComp . . . . .	2
DWLasso . . . . .	3
MBLasso . . . . .	5
weightComp . . . . .	6
weightEstim . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

combined	<i>Convert the list into a symmetric matrix</i>
----------	---

---

**Description**

This function converts the list of vectors into a symmetric matrix

**Usage**

```
combined(dat,y)
```

**Arguments**

dat	An input matrix. The columns represent variables and the rows indicate observations.
y	List data that includes estimates from nodewise regression.

**Value**

symMat	Symmetric matrix that represents the inferred network
--------	---

---

degreeComp	<i>Computing the degree of the inferred network</i>
------------	---

---

**Description**

This function computes the degree of inferred network

**Usage**

```
degreeComp(out.mat)
```

**Arguments**

out.mat	Symmetric matrix that represents the inferred network
---------	---

**Details**

This function computes the degree of estimated networks.

**Value**

wlasso_norm	Degree vector computed from the inferred network
-------------	--

**Examples**

```

library(DWLasso)
library(glmnet)
library(hglasso)

# Generate inverse covariance matrix with 3 hubs
# 20 % of the elements within a hub are zero
# 97 % of the elements that are not within hub nodes are zero
p <- 60 # Number of variables
n <- 40 # Number of samples

hub_number = 3 # Number of hubs

# Generate the adjacency matrix
Theta <- HubNetwork(p,0.97,hub_number,0.2)$Theta

# Generate a data matrix
out <- rmvnorm(n,rep(0,p),solve(Theta))

# Standardize the data
dat <- scale(out)

# Infer the network using weighted nodewise regression
w.mb <- rep(1,p)
adj.mat <- MBLasso(dat,lambda=0.4,w.mb)

# Compute the degree of the inferred network
deg.mat <- degreeComp(adj.mat)

```

---

DWLasso

*Degree weighted lasso*


---

**Description**

Infers undirected networks with hubs using weighted nodewise regression approach. The method contains two parameters that control hub and overall sparsity, respectively.

**Usage**

```
DWLasso(X, lambda1 = 0.4, lambda2 = 2, a = 1, tol = 1e-05)
```

**Arguments**

X	An input matrix. The columns represent variables and the rows indicate observations.
lambda1	A penalty parameter that controls degree sparsity of the inferred network
lambda2	A penalty parameter that controls overall sparsity of the inferred network
a	A parameter of the update equation that controls the convergence of weights
tol	Tolerance

## Details

This implements weighted degree lasso using coordinate descent algorithm (implemented in Glmnet package) described in Sulaimanov et al.. The method is based on the weighted nodewise regression approach and infers large undirected networks with hubs in iterative manner in the setting more variables than samples ( $p > n$ ). Given  $p$  variables, the network is inferred by regressing each variable against the remaining  $(p-1)$  variables. The penalty parameter,  $\lambda_1$  controls the degree sparsity of the network, whereas the penalty parameter,  $\lambda_2$  controls the overall sparsity. The method uses a fast Lasso solver Glmnet (Friedman et al. (2010)) with default settings.

## Value

mat	The estimated matrix corresponding to the inferred network. The diagonal elements of the matrix are zero
weights	The estimated weights used to estimate the network. These weights are computed from the degree of estimated networks
lambda1	The value of the penalty parameter controlling degree sparsity of the inferred network.
lambda2	The value of the penalty parameter controlling the overall sparsity

## Author(s)

Nurgazy Sulaimanov, Sunil Kumar, Frederic Burdet, Mark Ibberson, Marco Pagni, Heinz Koepl.

Maintainer: Nurgazy Sulaimanov, nurgazy.sulaimanov@bcs.tu-darmstadt.de

## References

1. Nurgazy Sulaimanov, Sunil Kumar, Frederic Burdet, Mark Ibberson, Marco Pagni, Heinz Koepl. Inferring hub networks using weighted degree Lasso. <http://arxiv.org/abs/1710.01912>.
2. Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1), 1-22. URL <http://www.jstatsoft.org/v33/i01/>.
3. Tan, KM., London, P., Mohan, K., Lee, S-I., Fazel, M., and Witten, D. (2014). Learning graphical models with hubs. *Journal of Machine Learning Research*. 5.1 (2014): 3297-3331.
4. Meinshausen, Nicolai, and Peter Bühlmann. "High-dimensional graphs and variable selection with the lasso." *The annals of statistics* (2006): 1436-1462.

## Examples

```
library(DWLasso)
library(glmnet)
library(hglasso)

# Generate inverse covariance matrix with 3 hubs
# 20 % of the elements within a hub are zero
# 97 % of the elements that are not within hub nodes are zero
p <- 60 # Number of variables
```

```
n <- 40 # Number of samples

hub_number = 3 # Number of hubs

# Generate the adjacency matrix
Theta <- HubNetwork(p,0.97,hub_number,0.2)$Theta

# Generate a data matrix
out <- rmvnorm(n,rep(0,p),solve(Theta))

# Standardize the data
dat <- scale(out)

# Run DWLasso
out.p <- DWLasso(dat, lambda1 = 0.6, lambda2 = 10)

# print out a summary of the output
summary(out.p)
```

---

MBLasso

*Inferring the network using nodewise regression method*

---

## Description

This function infers the network using nodewise regression method by Meinhausen and Buhlmann.

## Usage

```
MBLasso(dat, lambda=0.4, w.mb)
```

## Arguments

dat	An input matrix. The columns represent variables and the rows indicate observations.
lambda	A penalty parameter of the weighted Lasso that controls the sparsity of the inferred network.
w.mb	An unput weight vector which is computed from the degree of the inferred network.

## References

Meinshausen, Nicolai, and Peter Bühlmann. "High-dimensional graphs and variable selection with the lasso." *The annals of statistics* (2006): 1436-1462.

**Examples**

```

library(DWLasso)
library(glmnet)
library(hglasso)

# Generate inverse covariance matrix with 3 hubs
# 20 % of the elements within a hub are zero
# 97 % of the elements that are not within hub nodes are zero
p <- 60 # Number of variables
n <- 40 # Number of samples

hub_number = 3 # Number of hubs

# Generate the adjacency matrix
Theta <- HubNetwork(p,0.97,hub_number,0.2)$Theta

# Generate a data matrix
out <- rmvnorm(n,rep(0,p),solve(Theta))

# Standardize the data
dat <- scale(out)

# Infer the network using weighted nodewise regression
w.mb <- rep(1,p)
adj.mat <- MBLasso(dat,lambda=0.4,w.mb)

```

---

weightComp

*Computing weights from the degree of the inferred network*


---

**Description**

This function computes weights from the degree of estimated network using the weighted Lasso approach

**Usage**

```
weightComp(dat,lam=0.4,w.mb)
```

**Arguments**

dat	An input matrix. The columns represent variables and the rows indicate observations.
lam	A penalty parameter of the weighted Lasso that controls the sparsity of the inferred network.
w.mb	An unput weight vector which is computed from the degree of the inferred network.

**Value**

d.mb                    Weight vector computed from degree of the inferred network

**Examples**

```
library(DWLasso)
library(glmnet)
library(hglasso)

# Generate inverse covariance matrix with 3 hubs
# 20 % of the elements within a hub are zero
# 97 % of the elements that are not within hub nodes are zero
p <- 60 # Number of variables
n <- 40 # Number of samples

hub_number = 3 # Number of hubs

# Generate the adjacency matrix
Theta <- HubNetwork(p,0.97,hub_number,0.2)$Theta

# Generate a data matrix
out <- rmvnorm(n,rep(0,p),solve(Theta))

# Standardize the data
dat <- scale(out)

# Compute weights from the inferred network
w.mb <- rep(1,p)
w.Mat <- weightComp(dat,lam=0.4,w.mb)
```

---

weightEstim

*Estimating weights from the degree of the inferred network*


---

**Description**

This function estimates weights from the degree of the inferred network using iterative procedure. This function is called from the main function DWLasso.R

**Usage**

```
weightEstim(dat, lam=0.4, a=1, tol=1e-6)
```

**Arguments**

dat	An input matrix. The columns represent variables and the rows indicate observations.
lam	A penalty parameter that controls degree sparsity of the inferred network
a	A parameter of the update equation that controls the convergence of weights
tol	Tolerance

**Value**

w.dat                    Estimated weight vector from the last iteration at which the algorithm converges

**Author(s)**

Nurgazy Sulaimanov, Sunil Kumar and Heinz Koepl

**References**

Nurgazy Sulaimanov, Sunil Kumar, Frederic Burdet, Mark Ibberson, Marco Pagni, Heinz Koepl. Inferring hub networks using weighted degree Lasso. <http://arxiv.org/abs/1710.01912>.

**Examples**

```
library(DWLasso)
library(glmnet)
library(hglasso)

# Generate inverse covariance matrix with 3 hubs
# 20 % of the elements within a hub are zero
# 97 % of the elements that are not within hub nodes are zero
p <- 60 # Number of variables
n <- 40 # Number of samples

hub_number = 3 # Number of hubs

# Generate the adjacency matrix
Theta <- HubNetwork(p,0.97,hub_number,0.2)$Theta

# Generate a data matrix
out <- rmvnorm(n,rep(0,p),solve(Theta))

# Standardize the data
dat <- scale(out)

# Estimate weights from the degrees of the inferred network
w.est <- weightEstim(dat, lam=0.4, a=1, tol=1e-6)
```



# Index

combined, [2](#)

degreeComp, [2](#)

DWLasso, [3](#)

MBLasso, [5](#)

weightComp, [6](#)

weightEstim, [7](#)