# Package 'ADMMsigma'

March 21, 2018

**Type** Package

**Title** Penalized Precision Matrix Estimation via ADMM

**Version** 1.0

**Date** 2018-02-23

**Description** This R package produces penalized precision matrix estimates via the alternating direction method of multipliers (ADMM) algorithm

**License** MIT + file LICENSE

**ByteCompile** TRUE

**NeedsCompilation** yes

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Depends** Rcpp (>= 0.12.10),
RcppArmadillo,
doParallel,
foreach,
dplyr,
ggplot2

**LinkingTo** Rcpp,
RcppArmadillo

**Suggests** testthat

**SystemRequirements** GNU make

## R topics documented:

ADMMsigma *ADMM penalized precision matrix estimation (using ADMMsigmac)*

## Description

Penalized Gaussian likelihood precision matrix estimation using the ADMM algorithm.

## Usage

```
ADMMsigma(X = NULL, S = NULL, lam = 10^seq(-5, 5, 0.5), alpha = seq(0,
  1, 0.1), diagonal = FALSE, rho = 2, mu = 10, tau1 = 2, tau2 = 2,
  crit = "ADMM", tol1 = 1e-04, tol2 = 1e-04, maxit = 1000, K = 5,
  cores = 1, quiet = TRUE)
```

## Arguments

| | |
|---|---|
| X | data matrix |
| S | option to specify sample covariance matrix (denominator n) |
| lam | tuning parameter for penalty. Defaults to 10^seq(-5, 5, 0.5) |
| alpha | elasticnet mixing parameter [0, 1]: 0 = ridge, 1 = lasso/bridge. Defaults to seq(-1, 1, 0.1) |
| diagonal | option to penalize diagonal elements. Defaults to FALSE |
| rho | initial step size for ADMM |
| mu | factor for primal and residual norms |
| tau1 | adjustment for rho |
| tau2 | adjustment for rho |
| crit | criterion for convergence c('ADMM', 'grad', 'loglik'). Option crit != 'ADMM' will use tol1 as tolerance. Default is 'ADMM' |
| tol1 | absolute tolerance. Defaults to 1e-4 |
| tol2 | relative tolerance. Defaults to 1e-4 |
| maxit | maximum number of iterations |
| K | specify the number of folds for cross validation |
| cores | option to run CV in parallel. Defaults to cores = 1 |
| quiet | specify whether the function returns progress of CV or not |

## Value

iterations, lam, omega, and gradient

## Examples

```
ADMM_sigma(X, lam = 0.1, rho = 10)
```

## plot.ADMMsigma *Plot ADMMsigma object*

### Description

produces a heat plot for the cross validation errors

### Usage

```
## S3 method for class 'ADMMsigma'
plot(x, footnote = TRUE, ...)
```

### Arguments

| | |
|---|---|
| x | ADMMsigma class object |
| footnote | option to print footnote of optimal values |

## plot.RIDGEsigma *Plot RIDGEsigma object*

### Description

produces a heat plot for the cross validation errors

### Usage

```
## S3 method for class 'RIDGEsigma'
plot(x, footnote = TRUE, ...)
```

### Arguments

| | |
|---|---|
| x | RIDGEsigma class object |
| footnote | option to print footnote of optimal values |

## RIDGEsigma *Ridge penalized precision matrix estimation (using RIDGEsigmac)*

### Description

Penalized Gaussian likelihood precision matrix estimation using the ADMM algorithm.

### Usage

```
RIDGEsigma(X = NULL, S = NULL, lam = 10^seq(-5, 5, 0.5), K = 3,
  quiet = TRUE)
```

## Arguments

| | |
|---|---|
| X | data matrix |
| S | option to specify sample covariance matrix (denominator n) |
| lam | tuning parameter for penalty. Defaults to 10^seq(-5, 5, 0.5) |
| K | specify the number of folds for cross validation |
| quiet | specify whether the function returns progress of CV or not |

## Value

lam, omega, and gradient

## Examples

```
RIDGEsigma(X, lam = 0.1)
```

# Index