

# Package ‘ADMMsigma’

February 24, 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  
**Imports** Rcpp (>= 0.12.10),  
RcppArmadillo  
**LinkingTo** Rcpp,  
RcppArmadillo,  
RcppParallel  
**Suggests** testthat  
**SystemRequirements** GNU make

## R topics documented:

ADMMsigma . . . . .	2
ADMMsigmac . . . . .	3
CV_ADMMsigmac . . . . .	4
CV_sigma_ridgec . . . . .	5
<b>Index</b>	<b>6</b>

ADMMsigma

*ADMM penalized precision matrix estimation (using ADMM\_sigma)***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 = 1,
  rho = 2, mu = 10, tau1 = 2, tau2 = 2, crit = "ADMM", tol1 = 1e-04,
  tol2 = 1e-04, maxit = 1000, K = 3, quiet = TRUE)
```

**Arguments**

lam	tuning parameter for penalty. Defaults to $10^{\text{seq}(-5, 5, 0.5)}$
alpha	elasticnet mixing parameter [0, 1]: 0 = ridge, 1 = lasso/bridge
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', 'lik'). 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
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)
```

ADMMsigmac

*ADMM penalized precision matrix estimation (c++)***Description**

Penalized Gaussian likelihood precision matrix estimation using the ADMM algorithm.

**Usage**

```
ADMMsigmac(S, lam, alpha = 1, rho = 2, mu = 10, tau1 = 2, tau2 = 2,
  crit = "ADMM", tol1 = 1e-04, tol2 = 1e-04, maxit = 1000L)
```

**Arguments**

S	option to specify sample covariance matrix (denominator n)
lam	tuning parameter for penalty
alpha	elasticnet mixing parameter [0, 1]: 0 = ridge, 1 = lasso/bridge
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", "lik"). Option crit != "ADMM" will use tol1 as tolerance. Defaults to "ADMM"
tol1	absolute tolerance. Defaults to 1e-4
tol2	relative tolerance. Defaults to 1e-4
maxit	maximum number of iterations
X	data matrix

**Value**

iterations, lam, omega

**Examples**

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

---

CV_ADMMsigmac	<i>CV ADMM penalized precision matrix estimation (c++)</i>
---------------	--

---

## Description

Cross validation function for ADMM\_sigma.

## Usage

```
CV_ADMMsigmac(X, lam, alpha, rho = 2, mu = 10, tau1 = 2, tau2 = 2,
  crit = "ADMM", tol1 = 1e-04, tol2 = 1e-04, maxit = 1000L, K = 3L,
  quiet = TRUE)
```

## Arguments

X	matrix or data frame. This is the n x p column matrix where the rows are a realization of n independent copies of a p-variate random vector
lam	tuning parameter for penalty. Defaults to $10^{\text{seq}(-5, 5, 0.5)}$
alpha	elasticnet mixing parameter [0, 1]: 0 = ridge, 1 = lasso/bridge
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', 'lik'). Option crit != 'ADMM' will use tol1 as tolerance. Defaults to '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
quiet	specify whether the function returns progress of CV or not

## Value

iterations, lam, S, Omega, and cv.errors

## Examples

```
CV_ADMMsigmac(X, lam = seq(0.1, 3, 0.1))
```

---

CV_sigma_ridgec	<i>CV ADMM penalized precision matrix estimation (c++)</i>
-----------------	--

---

**Description**

Cross validation function for ADMM\_sigma.

**Usage**

```
CV_sigma_ridgec(X, lam, K = 3L, quiet = TRUE)
```

**Arguments**

X	matrix or data frame. This is the $n \times p$ column matrix where the rows are a realization of $n$ independent copies of a $p$ -variate random vector
lam	tuning parameter for penalty. Defaults to $10^{\text{seq}(-5, 5, 0.5)}$
K	specify the number of folds for cross validation
quiet	specify whether the function returns progress of CV or not

**Value**

iterations, lam, S, Omega, and cv.errors

**Examples**

```
CV_ADMMsigmac(X, lam = seq(0.1, 3, 0.1))
```

# Index

ADMMsigma, [2](#)

ADMMsigmac, [3](#)

CV\_ADMMsigmac, [4](#)

CV\_sigma\_ridgec, [5](#)