

Package ‘admmDensestSubmatrix’

July 22, 2025

Type Package

Title Alternating Direction Method of Multipliers to Solve Dense
Submatrix Problem

Version 0.1.0

Author Brendan Ames <bpames@ua.edu>, Polina Bombina <pbombina@crimson.ua.edu>

Maintainer Polina Bombina <pbombina@crimson.ua.edu>

Description Solves the problem of identifying the densest submatrix in a given or sampled binary matrix, Bombina et al. (2019) <[doi:10.48550/arXiv.1904.03272](https://doi.org/10.48550/arXiv.1904.03272)>.

License CC0

Depends R (>= 3.5.0)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports Rdpack, utils, stats

RdMacros Rdpack

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-31 16:20:02 UTC

Contents

densub	2
mat_shrink	3
plantedsubmatrix	3

Index	5
--------------	----------

densub

*densub***Description**

Iteratively solves the convex optimization problem using ADMM.

Usage

```
densub(G, m, n, tau = 0.35, gamma = 6/(sqrt(m * n) * (q - p)),
      opt_tol = 1e-04, maxiter, quiet = TRUE)
```

Arguments

G	sampled binary matrix
m	number of rows in dense submatrix
n	number of columns in dense submatrix
tau	penalty parameter for equality constraint violation
gamma	l_1 regularization parameter
opt_tol	stopping tolerance in algorithm
maxiter	maximum number of iterations of the algorithm to run
quiet	toggles between displaying intermediate statistics

Details

$$\min |X|_* + \gamma |Y|_1 + 1_{\Omega_W}(W) + 1_{\Omega_Q}(Q) + 1_{\Omega_Z}(Z)$$

$$\text{s.t } X - Y = 0, X = W, X = Z,$$

where $\Omega_W(W)$, $\Omega_Q(Q)$, $\Omega_Z(Z)$ are the sets: $\Omega_W = \{W \in \mathbb{R}^{M \times N} | e^T W e = mn\}$

$$\Omega_Q = \{Q \in \mathbb{R}^{M \times N} | \text{Projection of } Q \text{ on } N = 0\}$$

$$\Omega_Z = \{Z \in \mathbb{R}^{M \times N} | Z_{ij} \leq 1 \text{ for all } (i, j) \in M \times N\}$$

$$\Omega_Q = \{Q \in \mathbb{R}^{M \times N} | \text{Projection of } Q \text{ on } N = 0\}$$

$$\Omega_Z = \{Z \in \mathbb{R}^{M \times N} | Z_{ij} \leq 1 \text{ for all } (i, j) \in M \times N\}$$

1_S is the indicator function of the set S in $\mathbb{R}^{M \times N}$ such that $1_S(X) = 0$ if X in S and +infinity otherwise

Value

Rank one matrix with mn nonzero entries, matrix Y that is used to count the number of disagreements between G and X

mat_shrink	<i>Soft thresholding operator.</i>
------------	------------------------------------

Description

Applies the shrinkage operator for singular value tresholding.

Usage

mat_shrink(K, tau)

Arguments

K	matrix
tau	regularization parameter

Value

Matrix

Examples

```
mat_shrink(matrix(c(1,0,0,0,1,1,1,1,1), nrow=3, ncol=3, byrow=TRUE),0.35)
```

plantedsubmatrix	<i>Sample matrix</i>
------------------	----------------------

Description

Generates binary (M, N) - matrix sampled from dense (m, n) - submatrix.

Usage

plantedsubmatrix(M, N, m, n, p, q)

Arguments

M	number of rows in sampled matrix
N	number of rows in sampled matrix
m	number of rows in dense submatrix
n	natural number used to calculate number of rows in dense submatrix
p	density outside planted submatrix
q	density inside planted submatrix

Details

Let U^* and V^* be m and n index sets. For each i in U^* , j in V^* we let $a_{ij} = 1$ with probability q and 0 otherwise. For each remaining ij we set $a_{ij} = 1$ with probability $p < q$ and take $a_{ij} = 0$ otherwise.

Value

Matrix G sampled from the planted dense (mn) -submatrix model, dense submatrix X_0 , matrix Y_0 used to count the number of disagreements between G and X_0

Examples

```
plantedsubmatrix(10,10,1,2,0.25,0.75)
```

Index

`densub`, [2](#)

`mat_shrink`, [3](#)

`plantedsubmatrix`, [3](#)