

Package ‘sparseLRMatrix’

May 9, 2026

Title Represent and Use Sparse + Low Rank Matrices

Version 0.1.0

Description Provides an S4 class for representing and interacting with sparse plus rank matrices. At the moment the implementation is quite spare, but the plan is eventually subclass Matrix objects.

License MIT + file LICENSE

URL <https://rohelab.github.io/sparseLRMatrix/>,
<https://github.com/RoheLab/sparseLRMatrix>

BugReports <https://github.com/RoheLab/sparseLRMatrix/issues>

Depends Matrix, methods

Imports RSpectra

Suggests covr, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1.9000

NeedsCompilation no

Author Alex Hayes [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-4985-5160>>)

Maintainer Alex Hayes <alexpgghayes@gmail.com>

Repository CRAN

Date/Publication 2021-03-02 10:50:02 UTC

Contents

dim,sparseLRMatrix-method	2
sparseLRMatrix	3
sparseLRMatrix-class	4
svds.sparseLRMatrix	4

dim,sparseLRMatrix-method

Check the dimension of a sparseLRMatrix

Description

Check the dimension of a sparseLRMatrix

Usage

```
## S4 method for signature 'sparseLRMatrix'  
dim(x)
```

Arguments

x A [sparseLRMatrix](#) object.

Value

Dimension of x.

Examples

```
set.seed(528491)  
  
n <- 50  
m <- 40  
k <- 3  
  
A <- rsparsematrix(n, m, 0.1)  
  
U <- Matrix(rnorm(n * k), nrow = n, ncol = k)  
V <- Matrix(rnorm(m * k), nrow = m, ncol = k)  
  
# construct the matrix, which represents A + U %*% t(V)  
X <- sparseLRMatrix(sparse = A, U = U, V = V)  
  
dim(X)  
  
s <- svds(X, 5) # efficient
```

sparseLRMatrix *Create a sparse plus low rank matrix*

Description

Create a sparse plus low rank matrix

Usage

```
sparseLRMatrix(sparse, U, V)
```

Arguments

sparse	sparseMatrix.
U	Matrix.
V	Matrix.

Value

A [sparseLRMatrix](#) S4 object.

Examples

```
set.seed(528491)

n <- 50
m <- 40
k <- 3

A <- rsparsematrix(n, m, 0.1)

U <- Matrix(rnorm(n * k), nrow = n, ncol = k)
V <- Matrix(rnorm(m * k), nrow = m, ncol = k)

# construct the matrix, which represents A + U %*% t(V)
X <- sparseLRMatrix(sparse = A, U = U, V = V)

dim(X)

s <- svds(X, 5) # efficient
```

sparseLRMatrix-class *Sparse plus low rank matrix*

Description

Eventually this class will subclass `Matrix` objects, but for now this is a basic implementation that essentially only supports singular value decomposition.

Details

To learn more about S4 classes, please see <https://adv-r.hadley.nz/s4.html>.

Slots

sparse sparseMatrix.

U Matrix.

V Matrix.

Examples

```
set.seed(528491)

n <- 50
m <- 40
k <- 3

A <- rsparsematrix(n, m, 0.1)

U <- Matrix(rnorm(n * k), nrow = n, ncol = k)
V <- Matrix(rnorm(m * k), nrow = m, ncol = k)

# construct the matrix, which represents A + U %*% t(V)
X <- sparseLRMatrix(sparse = A, U = U, V = V)

dim(X)

s <- svds(X, 5) # efficient
```

svds.sparseLRMatrix *Truncated singular value decomposition of a matrix*

Description

A thin wrapper around `RSpectra::svds()`, please see more detailed documentation there. In particular, this function leverages the function interface.

Usage

```
## S3 method for class 'sparseLRMatrix'  
svds(A, k, nu = k, nv = k, opts = list(), ...)
```

Arguments

A	Matrix to decompose.
k	Number of singular values to estimate.
nu	Number of left singular vectors to estimate.
nv	Number of right singular vectors to estimate.
opts	Passed to <code>RSpectra::svds()</code> .
...	Passed to <code>RSpectra::svds()</code> .

Value

A list with the following components:

d	A vector of the computed singular values.
u	An m by nu matrix whose columns contain the left singular vectors. If $nu == 0$, NULL will be returned.
v	An n by nv matrix whose columns contain the right singular vectors. If $nv == 0$, NULL will be returned.
nconv	Number of converged singular values.
niter	Number of iterations used.
nops	Number of matrix-vector multiplications used.

Examples

```
set.seed(528491)  
  
n <- 50  
m <- 40  
k <- 3  
  
A <- rsparsematrix(n, m, 0.1)  
  
U <- Matrix(rnorm(n * k), nrow = n, ncol = k)  
V <- Matrix(rnorm(m * k), nrow = m, ncol = k)  
  
X <- sparseLRMatrix(sparse = A, U = U, V = V)  
  
svds(X, 5)
```

Index

`dim, sparseLRMatrix-method`, [2](#)

`RSpectra::svds()`, [4](#), [5](#)

`sparseLRMatrix`, [2](#), [3](#), [3](#)

`sparseLRMatrix-class`, [4](#)

`svds.sparseLRMatrix`, [4](#)