

Package: pcaone (via r-universe)

October 11, 2026

Title Fast and Accurate Randomized Singular Value Decomposition
Algorithms with 'PCAone'

Version 1.5.0

Date 2025-03-26

Description Fast and Accurate Randomized Singular Value Decomposition
(RSVD) methods proposed in the 'PCAone' paper by Li (2023)
<<https://genome.cshlp.org/content/33/9/1599>>.

License GPL (>= 3)

Depends R (>= 4.0.0)

Imports Rcpp, Matrix (>= 1.1-0), stats

LinkingTo RcppEigen (>= 0.3.4.0.0), Rcpp

SystemRequirements C++17, GNU make

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/Zilong-Li/PCAoneR>

BugReports <https://github.com/Zilong-Li/PCAoneR/issues>

Encoding UTF-8

Config/pak/sysreqs make

Repository <https://zilong-li.r-universe.dev>

Date/Publication 2025-05-18 07:31:54 UTC

RemoteUrl <https://github.com/zilong-li/pcaoner>

RemoteRef HEAD

RemoteSha e6df6248342ecaba2cdb0ec9d791c32433b644da

Contents

dashSVD	2
pcaone	3
popgen	6
winSVD	6
Index	8

dashSVD	<i>Dynamic shifted Randomized SVD</i>
---------	---------------------------------------

Description

dashSVD implements the Dynamic shifted Randomized SVD proposed by Feng et al. 2024

Usage

```
dashSVD(A, k, p = 10, s = 10)
```

Arguments

A	array_like; a real/complex (m, n) input matrix (or data frame) to be decomposed.
k	integer; the target rank for the low-rank decomposition. k should satisfy $k \ll \min(m, n)$.
p	integer, optional; number of additional power iterations (by default $p = 7$).
s	integer, optional; oversampling parameter (by default $s = 10$).

Author(s)

Zilong Li <zilong.dk@gmail.com>

References

- Feng, Xu and Yu, Wenjian and Xie, Yuyang and Tang, Jie. "Algorithm 1043: Faster Randomized SVD with Dynamic Shifts" (2024) doi:[10.1145/3660629](https://doi.org/10.1145/3660629).

Examples

```
mat <- matrix(rnorm(100*20000), 20000, 100)
res <- dashSVD(mat, k = 10)
str(res)
```

pcaone

Randomized Singular Value Decomposition Algorithm with Window-based Power Iterations from PCAone (Li et al 2022).

Description

The Randomized Singular Value Decomposition (RSVD) computes the near-optimal low-rank approximation of a rectangular matrix using a fast probabilistic algorithm.

Usage

```
pcaone(  
  A,  
  k = NULL,  
  p = 10,  
  s = 20,  
  method = "auto",  
  B = 64,  
  shuffle = TRUE,  
  opts = list()  
)  
  
## S3 method for class 'matrix'  
pcaone(  
  A,  
  k = NULL,  
  p = 10,  
  s = 20,  
  method = "auto",  
  B = 64,  
  shuffle = TRUE,  
  opts = list()  
)  
  
## S3 method for class 'dgCMatrix'  
pcaone(  
  A,  
  k = NULL,  
  p = 10,  
  s = 20,  
  method = "auto",  
  B = 64,  
  shuffle = TRUE,  
  opts = list()  
)  
  
## S3 method for class 'dgRMatrix'
```

```

pcaone(
  A,
  k = NULL,
  p = 7,
  s = 10,
  method = "winsvd",
  B = 64,
  shuffle = TRUE,
  opts = list()
)

```

Arguments

A	array_like; a real/complex (m, n) input matrix (or data frame) to be decomposed.
k	integer; the target rank for the low-rank decomposition. k should satisfy $k < \min(m, n)$.
p	integer, optional; number of power iterations (by default $p = 8$).
s	integer, optional; oversampling parameter (by default $s = 10$).
method	string $c('winsvd', 'dashsvd')$, optional; specifies the different variation of the randomized singular value decomposition : 'auto' (default): automatically choose the method based on size of input matrix. 'winsvd' : window based RSVD in PCAone paper. 'dashsvd' : dynamic shifts based RSVD with by Feng et al. 2024.
B	integer, optional; the number of batches for 'winsvd' method. must be a power of 2 (by default $B = 64$).
shuffle	logical, optional; if shuffle the rows of input tall matrix or not for winsvd (by default $shuffle = TRUE$).
opts	list, optional; options related to PCA, e.g. center and scale. See details

Details

The singular value decomposition (SVD) plays an important role in data analysis, and scientific computing. Given a rectangular (m, n) matrix A , and a target rank $k < \min(m, n)$, the SVD factors the input matrix A as

$$A = U_k \text{diag}(d_k) V_k^T$$

The k left singular vectors are the columns of the real or complex unitary matrix U . The k right singular vectors are the columns of the real or complex unitary matrix V . The k dominant singular values are the entries of d , and non-negative and real numbers.

q is an oversampling parameter to improve the approximation. A value of at least 10 is recommended, and $q = 10$ is set by default.

The parameter p specifies the number of power (subspace) iterations to reduce the approximation error. The power scheme is recommended, especially when the singular values decay slowly. However, computing power iterations increases the computational costs. Even though most RSVD implementations recommend $p = 3$ power iterations by default, it's always sufficient to run only few power iterations where our window-based power iterations ('alg2') come to play. We recommend using $windows = 64$ and $p \geq 7$ for `pcaone` algorithm2. As it is designed for large dataset, we recommend using 'alg2' when $\max(n, m) > 5000$.

If $k > (\min(n, m)/4)$, a deterministic partial or truncated `svd` algorithm might be faster.

Value

`pcaone` returns a list containing the following three components:

- d** array_like;
singular values; vector of length (k).
- u** array_like;
left singular vectors; (m, k) or (m, nu) dimensional array.
- v** array_like;
right singular vectors; (n, k) or (n, nv) dimensional array.

Note

The singular vectors are not unique and only defined up to sign. If a left singular vector has its sign changed, changing the sign of the corresponding right vector gives an equivalent decomposition.

Author(s)

Zilong Li <zilong.dk@gmail.com>

References

- Z. Li, J Meisner, A Albrechtsen. "Fast and accurate out-of-core PCA framework for large scale biobank data" (2023) [doi:10.1101/gr.277525.122](https://doi.org/10.1101/gr.277525.122).

Examples

```
library('pcaone')
load(system.file("extdata", "popgen.rda", package="pcaone"))
A <- popgen - rowMeans(popgen)
res <- pcaone(A, k = 40)
str(res)
```

popgen	<i>Population genetic data</i>
--------	--------------------------------

Description

Four populations from East Asia in the 1000 Genome Project

Format

An object of class `pcaone`.

Source

<https://www.internationalgenome.org>

winSVD	<i>Window-based Randomized SVD</i>
--------	------------------------------------

Description

winSVD implements the window-based Randomized SVD proposed by Li et al. 2023

Usage

```
winSVD(A, k, p = 10, s = 10, B = 64, perm = 1)
```

Arguments

A	array_like; a real (m, n) input matrix to be decomposed.
k	integer; specifies the target rank of the low-rank decomposition. k should satisfy $k \ll \min(m, n)$.
p	integer, optional; number of additional power iterations (by default $p = 10$).
s	integer, optional; oversampling parameter (by default $s = 10$).
B	integer, optional; number of windows (by default $B = 64$).
perm	integer, optional; the strategy for pivoting/permutation of rows : '1' : random (default) '2' : partial random '0' : no permutation

Value

winSVD returns a list containing the following three components:

- d** array_like;
singular values; vector of length (k) .
- u** array_like;
left singular vectors; (m, k) or (m, nu) dimensional array.
- v** array_like;
right singular vectors; (n, k) or (n, nv) dimensional array.

Note

The singular vectors are not unique and only defined up to sign. If a left singular vector has its sign changed, changing the sign of the corresponding right vector gives an equivalent decomposition.

Author(s)

Zilong Li <zilong.dk@gmail.com>

References

- Z. Li, J Meisner, A Albrechtsen. "Fast and accurate out-of-core PCA framework for large scale biobank data" (2023) [doi:10.1101/gr.277525.122](https://doi.org/10.1101/gr.277525.122).

Examples

```
mat <- matrix(rnorm(100*20000), 20000, 100)
res <- winSVD(mat, k = 10)
str(res)
```

Index

* **datasets**

popgen, [6](#)

dashSVD, [2](#)

pcaone, [3](#), [6](#)

popgen, [6](#)

svd, [5](#)

winSVD, [6](#)