Package: psvd (via r-universe)

October 26, 2024

Version 0.1-0

Date 2024-10-02

Title Eigendecomposition, Singular-Values and the Power Method

Author Doulaye Dembele [aut, cre]
(<https://orcid.org/0000-0003-3879-6940>)

Maintainer Doulaye Dembele <doulaye@igbmc.fr>

Depends R (>= 4.0),

Description For a data matrix with m rows and n columns (m>=n), the power method is used to compute, simultaneously, the eigendecomposition of a square symmetric matrix. This result is used to obtain the singular value decomposition (SVD) and the principal component analysis (PCA) results. Compared to the classical SVD method, the first r singular values can be computed.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-10-25 08:10:05 UTC

Contents

psvd-package *Eigendecomposition, Singular-Values and the Power Method*

Description

The power method is used to compute simultaneously the eigenvectors of a square symmetric matrix. Using the classical method, all eigenvectors are computed. The method used here allows to compute the first r eigenvectors using only matrix multiplications and the Gram-Schmidt orthogonalization algorithm. The relationships between the eigendecomposition factors, on the one hand, and the PCA factors or SVD factors, on the order hand, are used to get SVD or PCA results).

Details

Package:	psvd
Type:	Package
Version:	0.1-0
Date:	2024-10-02
License:	GPL (>= 2)

Package psvd has the following functions:

calcSVD():	Given a data matrix X of size (m,n), m >=n, this function allows to compute the singular value decomposition.
calcPCA():	Given a data matrix X of size (m,n), m >=n, this function allows to\ compute the principal component analysis.
mGS():	Modified Gramf-Schmidt orthogonalization method, R code, internal use.
mGSc():	Modified Gramf-Schmidt orthogonalization method, C code, internal use.
eigenV():	Computation of the eigenvectors matrix for a symmetric square matrix using the power method, R Code, internal use.
eigenVc():	Computation of the eigenvectors matrix for a symmetric square matrix using the power method, C Code, internal use.

Author(s)

Doulaye Dembele: <doulaye@igbmc.fr>

References

Dembele D. (2024), *Manuscript in preparation*

Examples

```
data(iris)
X <- as.matrix(iris[,1:4])
rownames(X) <- iris[,5]
res <- calcSVD(X, r=4)
```

```
res$d
res$v
res$iter
```

---

calcPCA *Perform principal component analysis*

Description

Given a data matrix, the function allows to perform principal component analysis using a power method to get the eigendecomposition.

Usage

```
calcPCA(X, r, eta, itmax, err, normed, mySeed)
```

Arguments

`X`	Data matrix of size (m,n), m >= n.
`r`	Number of principal components, default: r=2.
`eta`	Power method tuning parameter, default: eta=10.
`itmax`	Maximum number of iteration in the power method, default: itmax=200.
`err`	Tolerance level in the power method, default: err=1e-8.
`normed`	TRUE (default) or FALSE for PCA using standardized data or not.
`mySeed`	An integer allowing to reproduce results from two different runs, default: mySeed=50.

Details

X is usually a data matrix .

Value

This function returns a data frame containing 5 components

`values`	Eigenvalues
`vectors`	Matrix with the eigenvectors.
`iter`	The number of iterations used in the eigendecomposition.
`li`	Projection of rows in the r principal components space.
`co`	Projection of columns in the r principal components space.

Examples

```
data(iris)
X <- as.matrix(iris[,1:4])
rownames(X) <- iris[,5]
res <- calcPCA(X, r=3)
res$values
pcol <- c(rep("cyan",50), rep("red",50), rep("blue",50))
plot(res$li[,1], res$li[,3], col = pcol)
```

`calcSVD` *Perform singular values decomposition*

Description

Given a data matrix, the function allows to perform a singular decomposition using a power method and relationship between SVD factors and the eigendecomposition factors.

Usage

```
calcSVD(X, r, eta, itmax, err,mySeed)
```

Arguments

`X`	Data matrix of size (m,n), m >= n.
`r`	Rank r approximation, default: r=2.
`eta`	Power method tuning parameter, default: eta=10.
`itmax`	Maximum number of iteration in the power method, default: itmax=200.
`err`	Tolerance level in the power method, default: err=1e-8.
`mySeed`	An integer allowing to reproduce results from two different runs, default: mySeed=50.

Details

X is usually a data matrix.

Value

This function returns a data frame containing 4 components

`d`	Singular values.
`u`	Matrix with the right eigenvectors.
`v`	Matrix with the right eigenvectors.
`iter`	The number of iterations used in the eigendecomposition.

Examples

```
data(iris)
X <- as.matrix(iris[,1:4])
rownames(X) <- iris[,5]
res <- calcSVD(X, r=3)
res$d
res$v
res$iter
```

`eigenV` *Compute the eigenvectors matrix of a square symmetric matrix*

Description

This is an internal function which uses a R code to calculate an eidendecomposition of a square symmetric matrix. This function is used in the power method allowing to compute singular values and principal component analysis.

Usage

```
eigenV(xmat, wp, itmax, err)
```

Arguments

`xmat`	Square symmetric matrix of order d.
`wp`	Columns orthogonal matrix of size (d,r), r <= d.
`itmax`	Maximum number of iterations.
`err`	Tolerance level in the iterative search.

Value

This function returns a data frame containing 2 components

`wc`	Eigenvectors matrix.
`iter`	Number of iterations by the power method.

Examples

```
d <- 3
w <- matrix(rnorm(d*d,0,1), ncol=d)
wp <- mGS(w)
XtX <- matrix(c(3,2,1,2,1,0,1,0,1), ncol=3)
res <- eigenV(XtX, wp, itmax=100, err=1e-8)
t(res$wc)
```

`eigenVc` *Compute the eigenvectors of a square symmetric matrix*

Description

This is an internal function which uses a C code to calculate an eidendecomposition of a square symmetric matrix. This function is used in the power method allowing to compute singular values and principal component analysis.

Usage

```
eigenVc(xmat, wp, d, r, itmax, err)
```

Arguments

`xmat`	Square symmetric matrix of order d.
`wp`	Columns orthogonal matrix of size (d,r), r <= d.
`d`	Number of rows of wp.
`r`	Number of columns of wp.
`itmax`	Maximum number of iterations.
`err`	Tolerance level in the iterative search.

Value

This function returns a data frame containing 2 components

`wc`	Eigenvectors matrix.
`iter`	Number of iterations by the power method.

Examples

```
d <- 3
r <- 3
w <- c(rnorm(d*r,0,1))
res <- mGSc(w, d, r)
wp <- res$wp
XtX <- c(3,2,1,2,1,0,1,0,1)
res <- eigenVc(XtX, wp, d, r, itmax=100, err=1e-8)
wc <- matrix(res$wc, d, r)
t(wc)
```

mGS — *Modified Gram-Schmidt orthogonalization of a matrix*

Description

This is an internal function which uses a R code to calculate an orthogonalization of a matrix. This function is used in the power method allowing to compute an eigendecomposition of a symmetric square.

Usage

```
mGS(A)
```

Arguments

A	Matrix in vector form.

Value

This function returns a matrix which columns are the eigenvectors

wp	Eigevectors matrix.

Examples

```
A <- matrix(rnorm(6,0,1), ncol=2)
res <- mGS(A)
t(res)
```

mGSc — *Modified Gram-Schmidt orthogonalization of a matrix*

Description

This is an internal function which uses a C code to calculate an orthogonalization of a matrix. This function is used in the power method allowing to compute an eigendecomposition of a symmetric square.

Usage

```
mGSc(amat, m, n)
```

Arguments

amat	Matrix in vector form.
m	Number of rows of the matrix amat.
n	Number of columns of the matrix amat.

Value

This function returns a data frame containing 1 component

wp Eigenvectors matrix.

Examples

```
d <- 3
r <- 2
amat <- c(rnorm(d*r,0,1))
res <- mGSc(amat, d, r)
wp <- matrix(res$wp, nrow=d, ncol=r)
t(wp)
```

Index