Package ‘primePCA’

May 9, 2026

Type Package

Title Projected Refinement for Imputation of Missing Entries in PCA

Version 1.2

Date 2021-8-5

Author Ziwei Zhu, Tengyao Wang, Richard J. Samworth

Maintainer Ziwei Zhu <`ziweiz@umich.edu`>

Description Implements the primePCA algorithm, developed and analysed in Zhu, Z., Wang, T. and Samworth, R. J. (2019) High-dimensional principal component analysis with heterogeneous missingness. <doi:10.48550/arXiv.1906.12125>.

Imports softImpute, Matrix, MASS, methods

RoxygenNote 7.1.1

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2021-08-05 15:10:02 UTC

Contents

`col_scale` *Center and/or normalize each column of a matrix*

Description

Center and/or normalize each column of a matrix

Usage

```
col_scale(X, center = T, normalize = F)
```

Arguments

`X`	a numeric matrix with NAs or "Incomplete" matrix object (see softImpute package)
`center`	center each column of `X` if `center` == `TRUE`. The default value is `TRUE`.
`normalize`	normalize each column of `X` such that its sample variance is 1 if `normalize` == `TRUE`. The default value is `False`.

Value

a centered and/or normalized matrix of the same dimension as X.

`inverse_prob_method` *Inverse probability weighted method for estimating the top K eigenspaces*

Description

Inverse probability weighted method for estimating the top K eigenspaces

Usage

```
inverse_prob_method(X, K, trace.it = F, center = T, normalize = F)
```

Arguments

`X`	a numeric matrix with NAs or "Incomplete" matrix object (see softImpute package)
`K`	the number of principal components of interest
`trace.it`	report the progress if `trace.it` == `TRUE`
`center`	center each column of `X` if `center` == `TRUE`. The default value is `TRUE`.
`normalize`	normalize each column of `X` such that its sample variance is 1 if `normalize` == `TRUE`. The default value is `False`.

Value

Columnwise centered matrix of the same dimension as X.

Examples

```
X <- matrix(1:30 + .1 * rnorm(30), 10, 3)
X[1, 1] <- NA
X[2, 3] <- NA
v_hat <- inverse_prob_method(X, 1)
```

`primePCA` *primePCA algorithm*

Description

primePCA algorithm

Usage

```
primePCA(
  X,
  K,
  V_init = NULL,
  thresh_sigma = 10,
  max_iter = 1000,
  thresh_convergence = 1e-05,
  thresh_als = 1e-10,
  trace.it = F,
  prob = 1,
  save_file = "",
  center = T,
  normalize = F
)
```

Arguments

`X`	an n-by-d data matrix with `NA` values
`K`	the number of the principal components of interest
`V_init`	an initial estimate of the top K eigenspaces of the covariance matrix of `X`. By default, primePCA will be initialized by the inverse probability method.
`thresh_sigma`	used to select the "good" rows of X to update the principal eigenspaces σ_* in the paper).
`max_iter`	maximum number of iterations of refinement
`thresh_convergence`	The algorithm is halted if the Frobenius-norm sine-theta distance between the two consecutive iterates

`thresh_als`	This is fed into `thresh` in `svd.als` of `softImpute`. is less than `thresh_convergence`.
`trace.it`	report the progress if `trace.it` = `TRUE`
`prob`	probability of reserving the "good" rows. `prob` == `1` means to reserve all the "good" rows.
`save_file`	the location that saves the intermediate results, including `V_cur`, `step_cur` and `loss_all`, which are introduced in the section of returned values. The algorithm will not save any intermediate result if `save_file` == `""`.
`center`	center each column of `X` if `center` == `TRUE`. The default value is `TRUE`.
`normalize`	normalize each column of `X` such that its sample variance is 1 if `normalize` == `TRUE`. The default value is `False`.

Value

a list is returned, with components `V_cur`, `step_cur` and `loss_all`. `V_cur` is a d-by-K matrix of the top K eigenvectors. `step_cur` is the number of iterations. `loss_all` is an array of the trajectory of MSE.

Examples

```
X <- matrix(1:30 + .1 * rnorm(30), 10, 3)
X[1, 1] <- NA
X[2, 3] <- NA
v_tilde <- primePCA(X, 1)$V_cur
```

`sin_theta_distance` *Frobenius norm sin theta distance between two column spaces*

Description

Frobenius norm sin theta distance between two column spaces

Usage

```
sin_theta_distance(V1, V2)
```

Arguments

`V1`	a matrix with orthonormal columns
`V2`	a matrix of the same dimension as V1 with orthonormal columns

Value

the Frobenius norm sin theta distance between two V1 and V2

Index