Package ‘sparseEigen’

October 14, 2022

Title Computation of Sparse Eigenvectors of a Matrix

Version 0.1.0

Description Computation of sparse eigenvectors of a matrix (aka sparse PCA) with running time 2-3 orders of magnitude lower than existing methods and better final performance in terms of recovery of sparsity pattern and estimation of numerical values. Can handle covariance matrices as well as data matrices with real or complex-valued entries. Different levels of sparsity can be specified for each individual ordered eigenvector and the method is robust in parameter selection. See vignette for a detailed documentation and comparison, with several illustrative examples.
The package is based on the paper:
K. Benidis, Y. Sun, P. Babu, and D. P. Palomar (2016). ``Orthogonal Sparse PCA and Covariance Estimation via Procrustes Reformulation,'' IEEE Transactions on Signal Processing <doi:10.1109/TSP.2016.2605073>.

Maintainer Daniel P. Palomar <daniel.p.palomar@gmail.com>

URL https://github.com/dppalomar/sparseEigen,
https://www.danielppalomar.com,
https://doi.org/10.1109/TSP.2016.2605073

BugReports https://github.com/dppalomar/sparseEigen/issues

Depends R (>= 3.4.0)

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

Imports

RoxygenNote 6.0.1

Suggests knitr, MASS, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Konstantinos Benidis [aut],
Daniel P. Palomar [cre, aut]

Repository CRAN

Date/Publication 2017-12-21 16:54:39

R topics documented:

`spEigen` *Sparse Spectral Decomposition of a Matrix*

Description

Computes sparse (orthogonal) eigenvectors of covariance matrix or directly of data matrix.

Usage

```
spEigen(X, q = 1, rho = 0.5, data = FALSE, d = NA, V = NA,
  thres = 1e-09)
```

Arguments

`X`	m-by-m covariance matrix or n-by-m data matrix (n samples, m variables). Both real and complex matrices are accepted.
`q`	number of eigenvectors to be estimated.
`rho`	sparsity weight factor. Any nonnegative number (suggested range [0,1]).
`data`	boolean variable. If `TRUE`, `X` is treated as a data matrix, else as a covariance matrix (default).
`d`	vector with q weights. The default value is `seq(from = 1, to = 0.5, length.out = q)`.
`V`	initial m-by-q matrix point. If not provided, the eigenvectors of the sample covariance matrix are used.
`thres`	threshold value. All the entries of the sparse eigenvectors less or equal to `thres` are set to 0. The default value is `1e-9`.

Value

A list with the following components:

`vectors`	m-by-q matrix, columns corresponding to the q leading sparse eigenvectors.
`standard_vectors`	m-by-q matrix, columns corresponding to standard (non-sparse) leading eigenvectors.
`values`	vector with the q leading eigenvalues (in decreasing order).

Author(s)

Konstantinos Benidis and Daniel P. Palomar

References

K. Benidis, Y. Sun, P. Babu, D. P. Palomar, "Orthogonal Sparse PCA and Covariance Estimation via Procrustes Reformulation," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6211-6226, Dec. 2016.

Examples

```
library(sparseEigen)
n <- 100  # samples
m <- 500  # dimension
q <- 3  # number of sparse eigenvectors to be estimated
sp_card <- 0.1*m  # sparsity of each eigenvector

# generate covariance matrix with sparse eigenvectors
V <- matrix(0, m, q)
V[cbind(seq(1, q*sp_card), rep(1:q, each = sp_card))] <- 1/sqrt(sp_card)
V <- cbind(V, matrix(rnorm(m*(m-q)), m, m-q))
V <- qr.Q(qr(V))  # orthogonalize eigenvectors
lmd <- c(100*seq(from = q, to = 1), rep(1, m-q))  # generate eigenvalues
R <- V %*% diag(lmd) %*% t(V)  # covariance matrix

# generate data
X <- MASS::mvrnorm(n, rep(0, m), R)  # random data with underlying sparse structure

# standardand sparse eigenvectors
res_standard <- eigen(cov(X))
res_sparse <- spEigen(cov(X), q)

# show inner product between estimated eigenvectors and originals (the closer to 1 the better)
abs(diag(t(res_standard$vectors) %*% V[, 1:q]))  #for standard estimated eigenvectors
abs(diag(t(res_sparse$vectors) %*% V[, 1:q]))    #for sparse estimated eigenvectors
```

spEigenCov *Covariance Matrix Estimation with Sparse Eigenvectors*

Description

Estimates the covariance matrix with sparse (orthogonal) eigenvectors (in other words, it jointly estimates the sparse eigenvectors and the eigenvalues).

Usage

```
spEigenCov(S, q = 1, rho = 0.5, thres = 1e-09)
```

Arguments

S	m-by-m sample covariance matrix. It is required that S is full-rank. Both real and complex matrices are accepted.
q	number of sparse eigenvectors.
rho	sparsity weight factor. Any nonnegative number (suggested range [0,1]).
thres	threshold value. All the entries of the sparse eigenvectors less or equal to thres are set to 0. The default value is 1e-9.

Value

A list with the following components:

vectors	m-by-m matrix, columns corresponding to eigenvectors.
values	m-by-1 vector corresponding to eigenvalues.

Author(s)

Konstantinos Benidis and Daniel P. Palomar

References

K. Benidis, Y. Sun, P. Babu, D. P. Palomar, "Orthogonal Sparse PCA and Covariance Estimation via Procrustes Reformulation," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6211-6226, Dec. 2016.

Examples

```
## Not run:
library(sparseEigen)
n <- 600  # samples
m <- 500  # dimension
q <- 3  # number of sparse eigenvectors to be estimated
sp_card <- 0.1*m  # sparsity of each eigenvector

# generate covariance matrix with sparse eigenvectors
V <- matrix(0, m, q)
V[cbind(seq(1, q*sp_card), rep(1:q, each = sp_card))] <- 1/sqrt(sp_card)
V <- cbind(V, matrix(rnorm(m*(m-q)), m, m-q))
V <- qr.Q(qr(V))  # orthogonalize eigenvectors
lmd <- c(100*seq(from = q, to = 1), rep(1, m-q))  # generate eigenvalues
R <- V %*% diag(lmd) %*% t(V)  # covariance matrix

# generate data
X <- MASS::mvrnorm(n, rep(0, m), R)  # random data with underlying sparse structure

# standard and sparse estimation
res_standard <- eigen(cov(X))
res_sparse <- spEigenCov(cov(X), q)

# show inner product between estimated eigenvectors and originals (the closer to 1 the better)
```

```
abs(diag(t(res_standard$vectors) %*% V[, 1:q]))  #for standard estimated eigenvectors
abs(diag(t(res_sparse$vectors) %*% V[, 1:q]))    #for sparse estimated eigenvectors

# show error between estimated and true covariance
norm(cov(X) - R, type = 'F') #for sample covariance matrix
norm(res_sparse$cov - R, type = 'F') #for covariance with sparse eigenvectors

## End(Not run)
```

Index