Package ‘bdsvd’

January 8, 2025

Type Package

Title Block Structure Detection Using Singular Vectors

Version 0.2.1

Maintainer Jan O. Bauer <j.bauer@vu.nl>

Description Performs block diagonal covariance matrix detection using singular vectors (BD-SVD), which can be extended to hierarchical variable clustering (HC-SVD). The methods are described in Bauer (2024) <doi:10.1080/10618600.2024.2422985> and Bauer (202X) <doi:10.48550/arXiv.2308.06820>.

License GPL (>= 2)

Imports irlba, methods, stats

Encoding UTF-8

RoxygenNote 7.2.3

Suggests cvCovEst, mvtnorm, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Jan O. Bauer [aut, cre] (<https://orcid.org/0000-0001-7123-4507>),
Ron Holzapfel [aut]

Repository CRAN

Date/Publication 2025-01-08 11:10:08 UTC

Contents

bdsvd *Block Detection Using Singular Vectors (BD-SVD).*

Description

Performs BD-SVD iteratively to reveal the block structure. Splits the data matrix into one (i.e., no split) or two submatrices, depending on the structure of the first sparse loading v (which is a sparse approximation of the first right singular vector, i.e., a vector with many zero values) that mirrors the shape of the covariance matrix. This procedure is continued iteratively until the block diagonal structure has been revealed.

The data matrix ordered according to this revealed block diagonal structure can be obtained by `bdsvd.structure`.

Usage

```
bdsvd(X, dof.lim, anp = "2", standardize = TRUE, max.iter, trace = FALSE)
```

Arguments

`X`	Data matrix of dimension nxp with possibly $p >> n$.
`dof.lim`	Interval limits for the number of non-zero components in the sparse loading (degrees of freedom). If S denotes the support of v, then the cardinality of the support, $\|S\|$, corresponds to the degrees of freedom. Default is `dof.lim <- c(0, p-1)` which is highly recommended to check for all levels of sparsity.
`anp`	Which regularization function should be used for the HBIC. `anp = "1"` implements $a_{np} = 1$ which corresponds to the BIC, `anp = "2"` implements $a_{np} = 1/2log(np)$ which corresponds to the regularization used by Bauer (2024), and `anp = "3"` implements $a_{np} = log(log(np))$ which corresponds to the regularization used by Wang et al. (2009) and Wang et al. (2013).
`standardize`	Standardize the data to have unit variance. Default is `TRUE`.
`max.iter`	How many iterations should be performed for computing the sparse loading. Default is `200`.
`trace`	Print out progress as iterations are performed. Default is `TRUE`.

Details

The sparse loadings are computed using the method by Shen & Huang (2008), implemented by Baglama, Reichel, and Lewis in `ssvd` {irlba}.

Value

A list containing the feature names of the submatrices of `X`. The length of the list equals the number of submatrices.

References

Bauer, J.O. (2024). High-dimensional block diagonal covariance structure detection using singular vectors, J. Comput. Graph. Stat.

Wang, H., B. Li, and C. Leng (2009). Shrinkage tuning parameter selection with a diverging number of parameters, J. R. Stat. Soc. B 71 (3), 671–683.

Wang, L., Y. Kim, and R. Li (2013). Calibrating nonconvex penalized regression in ultra-high dimension, Ann. Stat. 41 (5), 2505–2536.

See Also

`bdsvd.structure`, `bdsvd.ht`, `single.bdsvd`

Examples

```
#Replicate the simulation study (c) from Bauer (2024).

## Not run:
p <- 500 #Number of variables
n <- 500 #Number of observations
b <- 10  #Number of blocks
design <- "c" #Simulation design "a", "b", "c", or "d".

#Simulate data matrix X
set.seed(1)
Sigma <- bdsvd.cov.sim(p = p, b = b, design = design)
X <- mvtnorm::rmvnorm(n, mean = rep(0, p), sigma = Sigma)
colnames(X) <- seq_len(p)

bdsvd(X, standardize = FALSE)

## End(Not run)
```

`bdsvd.cov.sim` *Covariance Matrix Simulation for BD-SVD*

Description

This function generates covariance matrices based on the simulation studies described in Bauer (2024).

Usage

```
bdsvd.cov.sim(p = p, b, design = design)
```

Arguments

`p`	Number of variables.
`b`	Number of blocks. Only required for simulation design "c" and "d".
`design`	Simulation design "a", "b", "c", or "d".

Value

A covariance matrix according to the chosen simulation design.

References

Bauer, J.O. (2024). High-dimensional block diagonal covariance structure detection using singular vectors, J. Comput. Graph. Stat.

Examples

```
#The covariance matrix for simulation design (a) is given by
Sigma <- bdsvd.cov.sim(p = 500, b = 500, design = "a")
```

`bdsvd.ht` *Hyperparameter Tuning for BD-SVD*

Description

Finds the number of non-zero elements of the sparse loading according to the high-dimensional Bayesian information criterion (HBIC).

Usage

```
bdsvd.ht(X, dof.lim, standardize = TRUE, anp = "2", max.iter)
```

Arguments

`X`	Data matrix of dimension nxp with possibly $p >> n$.
`dof.lim`	Interval limits for the number of non-zero components in the sparse loading (degrees of freedom). If S denotes the support of v, then the cardinality of the support, $\|S\|$, corresponds to the degrees of freedom. Default is `dof.lim <- c(0, p-1)` which is highly recommended to check for all levels of sparsity.
`standardize`	Standardize the data to have unit variance. Default is `TRUE`.
`anp`	Which regularization function should be used for the HBIC. `anp = "1"` implements $a_{np} = 1$ which corresponds to the BIC, `anp = "2"` implements $a_{np} = 1/2log(np)$ which corresponds to the regularization used by Bauer (2024), and `anp = "3"` implements $a_{np} = log(log(np))$ which corresponds to the regularization used by Wang et al. (2009) and Wang et al. (2013).
`max.iter`	How many iterations should be performed for computing the sparse loading. Default is `200`.

Details

The sparse loadings are computed using the method by Shen & Huang (2008), implemented in the `irlba` package. The computation of the HBIC is outlined in Bauer (2024).

Value

`dof`	The optimal number of nonzero components (degrees of freedom) according to the HBIC.
`BIC`	The HBIC for the different numbers of nonzero components.

References

Bauer, J.O. (2024). High-dimensional block diagonal covariance structure detection using singular vectors, J. Comput. Graph. Stat.

Shen, H. and Huang, J.Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation, J. Multivar. Anal. 99, 1015–1034.

Wang, H., B. Li, and C. Leng (2009). Shrinkage tuning parameter selection with a diverging number of parameters, J. R. Stat. Soc. B 71 (3), 671–683.

Wang, L., Y. Kim, and R. Li (2013). Calibrating nonconvex penalized regression in ultra-high dimension, Ann. Stat. 41 (5), 2505–2536.

See Also

`bdsvd`, `single.bdsvd`

Examples

```
#Replicate the illustrative example from Bauer (2024).


p <- 300 #Number of variables. In Bauer (2024), p = 3000
n <- 500 #Number of observations
b <- 3   #Number of blocks
design <- "c"

#Simulate data matrix X
set.seed(1)
Sigma <- bdsvd.cov.sim(p = p, b = b, design = design)
X <- mvtnorm::rmvnorm(n, mean = rep(0, p), sigma = Sigma)
colnames(X) <- seq_len(p)

ht <- bdsvd.ht(X)
plot(0:(p-1), ht$BIC[,1], xlab = "|S|", ylab = "HBIC", main = "", type = "l")
single.bdsvd(X, dof = ht$dof, standardize = FALSE)
```

bdsvd.structure — *Data Matrix Structure According to the Detected Block Structure.*

Description

Either sorts the data matrix X according to the detected block structure $X_1, ..., X_b$, ordered by the number of variables that the blocks contain. Or returns the detected submatrices each individually in a list object.

Usage

```
bdsvd.structure(X, block.structure, output = "matrix", block.order)
```

Arguments

`X`	Data matrix of dimension nxp with possibly $p >> n$.
`block.structure`	Output of `bdsvd()` or `single.bdsvd()` which identified the block structure.
`output`	Should the output be the data matrix ordered according to the blocks (`"matrix"`), or a list containing the submatrices (`"submatrices"`). Default is `"matrix"`.
`block.order`	A vector that contains the order of the blocks detected by `bdsvd()` or `single.bdsvd()`. The vector must contain the index of each blocks exactly once. Default is `1:b` where `b` is the total number of blocks.

Value

Either the data matrix `X` with columns sorted according to the detected blocks, or a list containing the detected submatrices.

References

Bauer, J.O. (2024). High-dimensional block diagonal covariance structure detection using singular vectors, J. Comput. Graph. Stat.

See Also

`bdsvd`, `single.bdsvd`

Examples

```
#Toying with the illustrative example from Bauer (2024).


p <- 150 #Number of variables. In Bauer (2024), p = 3000.
n <- 500 #Number of observations
b <- 3   #Number of blocks
design <- "c"
```

```
#Simulate data matrix X
set.seed(1)
Sigma <- bdsvd.cov.sim(p = p, b = b, design = design)
X <- mvtnorm::rmvnorm(n, mean = rep(0, p), sigma = Sigma)
colnames(X) <- seq_len(p)

#Compute iterative BD-SVD
bdsvd.obj <- bdsvd(X, standardize = FALSE)

#Obtain the data matrix X, sorted by the detected blocks
colnames(bdsvd.structure(X, bdsvd.obj, output = "matrix") )
colnames(bdsvd.structure(X, bdsvd.obj, output = "matrix", block.order = c(2,1,3)) )

#Obtain the detected submatrices X_1, X_2, and X_3
colnames(bdsvd.structure(X, bdsvd.obj, output = "submatrices")[[1]] )
colnames(bdsvd.structure(X, bdsvd.obj, output = "submatrices")[[2]] )
colnames(bdsvd.structure(X, bdsvd.obj, output = "submatrices")[[3]] )
```

---

block-class *Block*

Description

Class used within the package to store the structure and information about the detected blocks.

Slots

`features` numeric vector that contains the the variables corresponding to this block.

`block.columns` numeric vector that contains the indices of the singular vectors corresponding to this block.

---

detect.blocks *Block Detection*

Description

This function returns the block structure of a matrix.

Usage

```
detect.blocks(V, threshold = 0)
```

Arguments

`V`	Numeric matrix which either contains the loadings or is a covariance matrix.
`threshold`	All absolute values of `V` below the threshold are set to zero.

Value

An object of class `Block` containing the features and columns indices corresponding to each detected block.

References

Bauer, J.O. (2024). High-dimensional block diagonal covariance structure detection using singular vectors, J. Comput. Graph. Stat.

See Also

`bdsvd`, `single.bdsvd`

Examples

```
#In the first example, we replicate the simulation study for the ad hoc procedure
#Est_0.1 from Bauer (2024). In the second example, we manually compute the first step
#of BD-SVD, which can be done using the bdsvd() and/or single.bdsvd(), for constructed
#sparse loadings

#Example 1: Replicate the simulation study (a) from Bauer (2024) for the ad hoc
#procedure Est_0.1.

## Not run:
p <- 500 #Number of variables
n <- 125 #Number of observations
b <- 500 #Number of blocks
design <- "a"

#Simulate data matrix X
set.seed(1)
Sigma <- bdsvd.cov.sim(p = p, b = b, design = design)
X <- mvtnorm::rmvnorm(n, mean=rep(0, p), sigma=Sigma)
colnames(X) <- 1:p

#Perform the ad hoc procedure
detect.blocks(cvCovEst::scadEst(dat = X, lambda = 0.2), threshold = 0)

## End(Not run)

#Example 2: Manually compute the first step of BD-SVD
#for some loadings V that mirror the two blocks
#("A", "B") and c("C", "D").

V <- matrix(c(1,0,
              1,0,
              0,1,
              0,1), 4, 2, byrow = TRUE)

rownames(V) <- c("A", "B", "C", "D")
detected.blocks <- detect.blocks(V)
```

```
#Variables in block one with corresponding column index:
detected.blocks[[1]]@features
detected.blocks[[1]]@block.columns

#Variables in block two with corresponding column index:
detected.blocks[[2]]@features
detected.blocks[[2]]@block.columns
```

---

hcsvd *Hierarchical Variable Clustering Using Singular Vectors (HC-SVD).*

Description

Performs HC-SVD to reveal the hierarchical variable structure as descried in Bauer (202X). For this divise approach, each cluster is split into two clusters iteratively. Potential splits are identified by the first sparse loadings (which are sparse approximations of the first right eigenvectors, i.e., vectors with many zero values, of the correlation matrix) that mirror the masked shape of the correlation matrix. This procedure is continued until each variable lies in a single cluster.

Usage

```
hcsvd(
  R,
  q = "Kaiser",
  linkage = "average",
  is.corr = TRUE,
  max.iter,
  trace = TRUE
)
```

Arguments

`R`	A correlation matrix of dimension pxp or a data matrix of dimension nxp an be provided. If a data matrix is supplied, it must be indicated by setting `is.corr = FALSE`, and the correlation matrix will then be calculated as `cor(X)`.
`q`	Number of sparse loadings to be used. This should be either a numeric value between zero and one to indicate percentages, or `"Kaiser"` for as many sparse loadings as there are eigenvalues larger or equal to one. For a numerical value between zero and one, the number of sparse loadings is determined as the corresponding share of the total number of loadings. E.g., `q = 1` (100%) uses all sparse loadings and `q = 0.5` (50%) will use half of all sparse loadings.
`linkage`	The linkage function to be used. This should be one of `"average"`, `"single"`, or `"RV"` (for RV-coefficient).
`is.corr`	Is the supplied object a correlation matrix. Default is `TRUE` and this parameter must be set to `FALSE` is a data matrix instead of a correlation matrix is supplied.

max.iter	How many iterations should be performed for computing the sparse loadings. Default is 200.
trace	Print out progress as $p-1$ iterations for divisive hierarchical clustering are performed. Default is TRUE.

Details

The sparse loadings are computed using the method of Shen and Huang (2008), which is implemented based on the code of Baglama, Reichel, and Lewis in ssvd {irlba}, with slight modifications to suit our method.

Value

A list with four components:

hclust	The clustering structure identified by HC-SVD as an object of type hclust.
dist.matrix	The ultrametric distance matrix (cophenetic matrix) of the HC-SVD structure as an object of class dist.
u.cor	The ultrametric correlation matrix of X obtained by HC-SVD as an object of class matrix.
q.p	A vector of length $p-1$ containing the ratio q_i/p_i of the q_i sparse loadings used relative to all sparse loadings q_i for the split of each cluster. The ratio is set to NA if the cluster contains only two variables as the search for sparse loadings that reflect the split is not required in this case.

References

Bauer, J.O. (202X). Divisive hierarchical clustering identified by singular vectors.

Shen, H. and Huang, J.Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation, J. Multivar. Anal. 99, 1015–1034.

Examples

```
#We replicate the simulation study (a) in Bauer (202X)

## Not run:
p <- 40
n <- 500
b <- 5
design <- "a"

set.seed(1)
Rho <- hcsvd.cor.sim(p = p, b = b, design = "a")
X <- mvtnorm::rmvnorm(n, mean=rep(0, p), sigma = Rho, checkSymmetry = FALSE)
R <- cor(X)
hcsvd.obj <- hcsvd(R)

#The object of hclust with corresponding dendrogram can be obtained
#directly from hcsvd.obj$hclust:
hc <- hcsvd.obj$hclust
```

```
plot(hc)

#The dendrogram can also be obtained from the ultrametric distance matrix:
plot(hclust(hcsvd.obj$dist.matrix))

## End(Not run)
```

---

hcsvd.cor.sim *Correlation Matrix Simulation for HC-SVD*

Description

This function generates correlation matrices based on the simulation studies described in Bauer (202X).

Usage

```
hcsvd.cor.sim(p = p, b = b, design = design)
```

Arguments

p	Number of variables.
b	Number of blocks.
design	Simulation design "a" or "b".

Value

A correlation matrix according to the chosen simulation design.

References

Bauer, J.O. (202X). Divisive hierarchical clustering identified by singular vectors.

Examples

```
#The correlation matrix for simulation design (a) is given by
#R <- hcsvd.cov.sim(p = 40, b = 5, design = "a")
```

single.bdsvd *Single Iteration of Block Detection Using Singular Vectors (BD-SVD).*

Description

Performs a single iteration of BD-SVD: splits the data matrix into one (i.e., no split) or two submatrices, depending on the structure of the first sparse loading v (which is a sparse approximation of the first right singular vector, i.e., a vector with many zero values) that mirrors the shape of the covariance matrix.

Usage

```
single.bdsvd(X, dof, standardize = TRUE, max.iter)
```

Arguments

`X`	Data matrix of dimension nxp with possibly $p >> n$.
`dof`	Number of non-zero components in the sparse loading (degrees of freedom). If S denotes the support of v, then the cardinality of the support, $\|S\|$, corresponds to the degrees of freedom.
`standardize`	Standardize the data to have unit variance. Default is `TRUE`.
`max.iter`	How many iterations should be performed for computing the sparse loading. Default is `200`.

Details

The sparse loadings are computed using the method by Shen & Huang (2008), implemented in the `irlba` package.

Value

A list containing the feature names of the submatrices of `X`. It is either of length one (no split) or length two (split into two submatrices).

References

Bauer, J.O. (2024). High-dimensional block diagonal covariance structure detection using singular vectors, J. Comput. Graph. Stat.

Shen, H. and Huang, J.Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation, J. Multivar. Anal. 99, 1015–1034.

See Also

`bdsvd`, `bdsvd.ht`

Examples

```
#Replicate the illustrative example from Bauer (2024).

## Not run:

p <- 300 #Number of variables. In Bauer (2024), p = 3000.
n <- 500 #Number of observations
b <- 3   #Number of blocks
design <- "c"

#Simulate data matrix X
set.seed(1)
Sigma <- bdsvd.cov.sim(p = p, b = b, design = design)
X <- mvtnorm::rmvnorm(n, mean = rep(0, p), sigma = Sigma)
colnames(X) <- 1:p

ht <- bdsvd.ht(X)
plot(0:(p-1), ht$BIC[,1], xlab = "|S|", ylab = "HBIC", main = "", type = "l")
single.bdsvd(X, dof = ht$dof, standardize = FALSE)


## End(Not run)
```

Index