Package 'iSSVD'

May 16, 2024

Type Package

Title Intergrative Biclustering for Multi-view data with nested stability selection

Version 0.1.0

Author Jianfeng Wang, Weijie Zhang and Sandra Safo

Maintainer Sandra E. Safo <ssafo@umn.edu>

Description

The integrative sparse singular value decomposition (iSSVD) package implements the iSSVD algorithm for integrative biclustering of data from multiple sources. Biclustering refers to simultaneous clustering of samples and columns of a data matrix. This algorithm is useful for detecting sample subgroups characterized by specific groups of variables. iSSVD incorporates stability selection to control Type I error rates, estimates the probability of samples and variables to belong to a bicluster, finds stable biclusters, resulting in interpretable row-column associations.

License GPL (>=2.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Depends gtsummary, dplyr, rlang, purrr, ggplot2, reshape2, gridExtra, reticulate

R topics documented:

createVirtualenv *Create a Python virtual environment with necessary packages*

Description

This function is used to create a Python virtual environment (called "iSSVD") and installs necessary packages such as pandas, numpy, and scikit-learn to the environment. Once created, the environment will automatically be used upon loading the package.

Usage

```
createVirtualenv()
```

Details

If there is an error installing the Python packages, try restarting your computer and running R/RStudio as administrator. If the virtual environment is not created, the user's default system Python installation will be used. In this case, the user will need to have the following packages in their main local Python installation:

- pandas
- matplotlib
- seaborn
- scikit-learn
- numpy

Alternatively, the user can use their own virtual environment with reticulate by activating it with reticulate::use_virtualenv() or a similar function prior to loading RandMVLearn.

Examples

```
# Create Python virtual environment "iSSVD"
createVirtualenv()
```

diagnostics *Evaluating Biclustering Performance*

Description

The function evaluates the performance of biclustering results by comparing them against true biclusters using various metrics.

Usage

```
diagnostics(rows, cols, true_rows, true_cols, n = 200L, p = 1000L, D = 2L)
```

Arguments

rows	The indices of rows from the original dataset that have been grouped into biclusters by iSSVDR.
cols	A nested list of arrays corresponding to the column indices of the variables included in each bicluster for each view (dataset).
true_rows	The true indices of rows that define the biclusters in the original dataset. It is a list of arrays, each containing the row indices of a known true bicluster.
true_cols	A nested list of arrays containing the true indices of the variables that define the biclusters in each view of the original dataset.
n	The total number of samples (rows) in the original dataset. This is used to initialize arrays and to calculate the diagnostics. Need to append a letter L to the integer. Default: 200.
p	The total number of variables (columns) in the original dataset. Like n, this is used in the initialization of arrays for the diagnostics. Need to append a letter L to the integer. Default: 1000.
D	The number of views or datasets used in the integrative biclustering. Need to append a letter L to the integer. Default: 2.

Details

The function is designed to evaluate the performance of biclustering results, particularly focusing on the comparison between discovered biclusters and true biclusters. It calculates several diagnostic metrics, including recovery, relevance, F1 score, false positive rate, and false negative rate. Please refer to the main paper Zhang, Weijie, et al. "Robust integrative biclustering for multi-view data." Statistical methods in medical research 31.11 (2022): 2201-2216. for more details.

Value

A list with the following elements:

Recovery	Measures how well the detected clusters match the true clusters. A higher value indicates better recovery of the true cluster structure.
Relevance	Similar to recovery, this metric assesses the relevance of the detected clusters to the true clusters. A higher value indicates that the detected clusters are more relevant to the true cluster structure.
F-Score	The F-Score combines the information of both recovery and relevance into a single score, with higher values indicating better clustering performance.
False Positives (FPs)	The average minimum proportion of false positive elements across all detected clusters. A lower value indicates fewer elements are wrongly included, which is desirable.
False Negatives (FNs)	The average minimum proportion of false negative elements across all detected clusters. A lower value indicates fewer elements are wrongly excluded, which is also desirable.

References

Zhang, Weijie, et al. "Robust integrative biclustering for multi-view data." Statistical methods in medical research 31.11 (2022): 2201-2216.

Meinshausen, Nicolai, and Peter Bühlmann. "Stability selection." Journal of the Royal Statistical Society Series B: Statistical Methodology 72.4 (2010): 417-473.

Sill, Martin, et al. "Robust biclustering by sparse singular value decomposition incorporating stability selection." Bioinformatics 27.15 (2011): 2089-2097.

Examples

```
library(iSSVD)
# Generate two views with four biclusters, number of samples 50, number of variables 1000.
mydata = generateData(myseed = 20L) # Uses default settings
X1 = mydata[[1]][[1]][[1]]
X2 = mydata[[1]][[1]][[2]]
Xdata1 = list(X1, X2)

iSSVD_results = iSSVDR(X = Xdata1, myseed = 25L, standr = FALSE, vthr = 0.9, nbicluster = 4L)
sample_clusters = iSSVD_results[[3]][[3]]
variable_clusters = iSSVD_results[[3]][[4]]

true_sample_clusters = mydata[[2]][[1]]
true_variable_clusters = mydata[[3]][[1]]

diag = diagnostics(rows = sample_clusters, cols = variable_clusters, true_rows = true_sample_clusters, true_co
n = 200L, p = 1000L, D = 2L)
```

`generateData` *Generate simulated data for a given number of clusters*

Description

The function generate data matrices with the truth of underlying biclusters that can be used for dianostics.

Usage

```
generateData(
  myseed = 25L,
  n = 200L,
  p = 1000L,
  D = 2L,
  rowsize = 50L,
  colsize = 100L,
  numbers = 1L,
  sigma = 0.1,
  nbicluster = 4L,
  orthonm = FALSE
)
```

Arguments

`myseed`	An integer to set a seed. Need to append a letter L to the integer, for example 25L. It's set to a default value of 25.

n	An even integer for the number of rows in the generated data. It's set to a default value of 200.
p	An integer for the number of the number of columns in the generated data. It's set to a default value of 1000.
D	The number of views or data sources. In multi-view matrix factorization, you often have data from multiple sources or views. D controls how many such views are generated. It's set to a default value of 2.
rowsize	An integer for the number of rows in each submatrix. It's set to a default value of 50.
colsize	An integer for the number of columns in each submatrix. It's set to a default value of 100.
numbers	An integer for the number of sets of data matrices to be generated. It's set to a default value of 1.
sigma	A number for the standard deviation of the random noise added to the data. It's set to a default value of 0.1.
nbicluster	The number of clusters in the row and column spaces for each synthetic dataset. It's set to a default value of 4.
orthonm	A boolean flag. If TRUE, The left singular vector and right singular vector are orthogonalized. The default is FALSE.

Details

Please refer to the main paper Zhang, Weijie, et al. "Robust integrative biclustering for multi-view data." Statistical methods in medical research 31.11 (2022): 2201-2216. for more details.

Value

The function will return a list of synthetic data matrices. The list has a structure that reflects the different samples and factors in the generated data.

References

Zhang, Weijie, et al. "Robust integrative biclustering for multi-view data." Statistical methods in medical research 31.11 (2022): 2201-2216.

Meinshausen, Nicolai, and Peter Bühlmann. "Stability selection." Journal of the Royal Statistical Society Series B: Statistical Methodology 72.4 (2010): 417-473.

Sill, Martin, et al. "Robust biclustering by sparse singular value decomposition incorporating stability selection." Bioinformatics 27.15 (2011): 2089-2097.

Examples

```
# Generate the data list
mydata = generateData(myseed = 25L, n = 200L, p = 1000L, D = 2L, rowsize = 50L, colsize = 100L,
                      numbers = 1L, sigma = 0.1, nbicluster = 4L, orthonm = FALSE)

# Select two data X1, X2 from the generated data list
X1 = mydata[[1]][[1]][[1]]
X2 = mydata[[1]][[1]][[2]]

# Combine two data into a list
Xdata1 = list(X1, X2)
```

iSSVDR *Integrative Biclustering of Multi-view Data*

Description

The integrative sparse singular value decomposition (iSSVD) is based on sparse singular value decomposition for single-view data to data from multiple views. Theproposedalgorithmestimatestheprobabilityof samplesandvariables tobelongtoabicluster, finds stablebiclusters, andresults in interpretable row-column associations. Simulations and real data analyses show that iSSVD out performs several other single- and multi-view biclustering methods and is able to detect meaningful biclusters.

Usage

```
iSSVDR(
  X = Xdata,
  myseed = 25L,
  standr = FALSE,
  pointwise = TRUE,
  steps = 100L,
  size = 0.5,
  vthr = 0.9,
  ssthr = c(0.6, 0.65),
  nbicluster = 4L,
  rows_nc = FALSE,
  cols_nc = FALSE,
  col_overlap = FALSE,
  row_overlap = FALSE,
  pceru = 0.1,
  pcerv = 0.1,
  merr = 1e-04,
  iters = 100L,
  assign_unclustered_samples = TRUE
)
```

Arguments

X	Input data, expected to be a list of views or datasets that are integrated using the biclustering algorithm.
myseed	Seed for random number generation.
standr	A boolean flag for standardizing the data. Default: FALSE.
pointwise	If TRUE, a fast pointwise control method will be performed for stability selection. Default: TRUE.
steps	Number of subsamples used to perform stability selection. Default: 100.
size	Size of the subsamples used to perform stability selection. Default: 0.5.
vthr	Variance threshold for determining the initial estimate of the number of biclusters based on the cumulative sum of normalized singular values. Default: 0.9.
ssthr	Range of the threshold for stability selection. Default: c(0.6, 0.65).
nbicluster	A user specified number of biclusters to be detected. Default: 4.

`rows_nc` If TRUE allows for negative correlation of rows over columns. Default: FALSE.

`cols_nc` If TRUE allows for negative correlation of columns over rows. Default: FALSE.

`col_overlap` If TRUE allows for column overlaps among biclusters. Default: FALSE.

`row_overlap` If TRUE allows for row overlaps among biclusters. Default: FALSE.

`pceru` Per-comparison error rate to control the number of falsely selected coefficients in u. Default: 0.1.

`pcerv` Per-comparison error rate to control the number of falsely selected coefficients in v. Default: 0.1.

`merr` Minimum error threshold. Default: 0.0001.

`iters` Maximum iterations for detecting each bicluster. Default: 100.

`assign_unclustered_samples` If TRUE then automatically assigns unclustered samples to clusters based on the calculated probabilities indicating the likelihood of each unclustered sample belonging to a particular cluster. Default: TRUE.

Details

Suppose that data are available from D different views, and each view is arranged in an $n \times p^{(d)}$ matrix $X^{(d)}$, where the superscript d corresponds to the d-th view. For instance, for the same set of n individuals, matrix $X^{(1)}$ consists of RNA sequencing data and $X^{(2)}$ consists of proteomics data, for $D = 2$ views. We wish to cluster the $p^{(d)}$ variables, and the n subjects so that each subject subgroup is associated with a specific variable subgroup of relevant variables. We consider estimating the left and right singular vectors and inferring the non-zero coefficients, sample clusters, and variable clusters. We subsample variables in each view I times without replacement, while ensuring that each view contains the same set of samples. Please refer to the main paper Zhang, Weijie, et al. "Robust integrative biclustering for multi-view data." Statistical methods in medical research 31.11 (2022): 2201-2216. for more details. For this R manuscript, $X^{(d)}$, $X^{(1)}$, $X^{(2)}$ and $p^{(d)}$ is not showing as what I want

Value

A list with the following elements:

`results` Stability selection results of left and right singular vectors;

`N` Number of biclusters detected;

`Sample_index` The indices of bicluster samples;

`Variable_index` The indices of bicluster variables;

`Interaction` The interactions run for each bicluster;

`N_unclustered` Number of unclustered samples;

`unclustered_index` Indices of the samples that are not clustered;

`extract_data` The extracted data of each cluster and each view.

References

Zhang, Weijie, et al. "Robust integrative biclustering for multi-view data." Statistical methods in medical research 31.11 (2022): 2201-2216.

Meinshausen, Nicolai, and Peter Bühlmann. "Stability selection." Journal of the Royal Statistical Society Series B: Statistical Methodology 72.4 (2010): 417-473.

Sill, Martin, et al. "Robust biclustering by sparse singular value decomposition incorporating stability selection." Bioinformatics 27.15 (2011): 2089-2097.

Examples

```
library(iSSVD)
# Generate two views with four biclusters, number of samples 50, number of variables 1000.
mydata = generateData(myseed = 20L) # Uses default settings
X1 = mydata[[1]][[1]][[1]]
X2 = mydata[[1]][[1]][[2]]

Xdata1 = list(X1, X2)

iSSVD_results = iSSVDR(X = Xdata1, myseed = 25L, standr = FALSE, vthr = 0.9, nbicluster = 4L)

sample_clusters = lapply(iSSVD_results[[3]][[3]], as.integer)
variable_clusters = lapply(iSSVD_results[[3]][[4]], lapply, as.integer)

# Visualize cluster
plotHeatMapR(X = Xdata1, Rows = sample_clusters, Cols = variable_clusters, D = 2L, nbicluster = 4L)
```

`plotHeatMapR` *Plot heatmap using generated and results from iSSVDR*

Description

The plotHeatMapR function takes a data matrix, organizes it into biclusters, and creates and displays two side-by-side heatmaps to visualize the data's structure and relationships within two dimensions.

Usage

```
plotHeatMapR(X = Xdata, Rows = myRows, Cols = myCols, D = 2L, nbicluster = 4L)
```

Arguments

`X`	A list of matrices to be visualized. It's assumed that this list contains multiple views or data sources. The length of this list should be equal to the value of D, representing the number of views. Each matrix in the list corresponds to one view of the data.
`Rows`	A list of indices that represent the sample clusters from iSSVDR.
`Cols`	A list of indices that represent the variable clusters from iSSVDR.
`D`	This parameter represents the number of views or data sources in the synthetic data. It's set to 2 by default, but you can change it to match the actual number of views in your data.
`nbicluster`	The number of biclusters to visualize. This parameter specifies how many biclusters should be included in the heatmap visualization. It defaults to 4, but you can adjust it to visualize a different number of biclusters.

Details

Please refer to the main paper Zhang, Weijie, et al. "Robust integrative biclustering for multi-view data." Statistical methods in medical research 31.11 (2022): 2201-2216. for more details.

Value

The result of this function is the display of two heatmaps side by side, each representing a different dimension of the data.

References

Zhang, Weijie, et al. "Robust integrative biclustering for multi-view data." Statistical methods in medical research 31.11 (2022): 2201-2216.

Meinshausen, Nicolai, and Peter Bühlmann. "Stability selection." Journal of the Royal Statistical Society Series B: Statistical Methodology 72.4 (2010): 417-473.

Sill, Martin, et al. "Robust biclustering by sparse singular value decomposition incorporating stability selection." Bioinformatics 27.15 (2011): 2089-2097.

Examples

```
# Generate data and perform integrative biclustering using iSSVDR()
mydata = generateData(myseed = 20L) # Uses default settings
X1 = mydata[[1]][[1]][[1]]
X2 = mydata[[1]][[1]][[2]]
Xdata1 = list(X1, X2)

iSSVD_results = iSSVDR(X = Xdata1, myseed = 25L, standr = FALSE, vthr = 0.9, nbicluster = 4L)

# Plot HeatMap using the results from iSSVDR
myRows = iSSVD_results[[3]][[3]]
myCols = iSSVD_results[[3]][[4]]
myRows <- lapply(myRows, as.integer)
myCols <- lapply(myCols, lapply, as.integer)

plotHeatMapR(X = Xdata1, Rows = myRows, Cols = myCols, D = 2L, nbicluster = 4L)
```

---

summary_table *Make summary table for clinical data*

Description

The function allows the user to make a summary table based on clinical data and sample clusters get from iSSVDR() function.

Usage

```
summary_table(data, res, ...)
```

Arguments

`data`	Input clinical data. Can use data("clinical_data") to use example data that contains 120 rows and 4 variables: "Age", "Sex", "crp", "ddimer".
`res`	Input results directly generated from the iSSVDR() function.
`...`	Any variables you want to be shown in the summary table.

Details

summary_table returns a summary table showing you the summary statistics (N, Mean, SD) and p-value from Kruskal-Wallis rank sum test (for continuous variables)/ Pearson's Chi-squared test (for categorical variables). This function depends on the gtsummary package.

Value

A summary table that displays and compares summary statistics across sample clusters, and give p-values.

References

Zhang, Weijie, et al. "Robust integrative biclustering for multi-view data." Statistical methods in medical research 31.11 (2022): 2201-2216.

Meinshausen, Nicolai, and Peter Bühlmann. "Stability selection." Journal of the Royal Statistical Society Series B: Statistical Methodology 72.4 (2010): 417-473.

Sill, Martin, et al. "Robust biclustering by sparse singular value decomposition incorporating stability selection." Bioinformatics 27.15 (2011): 2089-2097.

Examples

```
library(iSSVD)
# Call example data
data("clinical_data") # Clinical data has no association with clusters, this is just for demonstrations.

# Generate three views with four biclusters, number of samples 120, number of variables 1000.
mydata = generateData(n = 120L, p = 1000L, D = 3L, rowsize = 30L, colsize = 100L,
                      numbers = 1L, sigma = 0.1, nbicluster = 4L, orthonm = FALSE)
Xdata1 <- list()
for (i in 1:3) {
  Xdata1[[i]] <- mydata[[1]][[1]][[i]]
}

res4 = iSSVDR(X = Xdata1, standr = FALSE, pointwise = TRUE, steps = 100, size = 0.5,
          vthr = 0.9, ssthr = c(0.6, 0.65), nbicluster = 4, rows_nc = FALSE, cols_nc = FALSE,
          col_overlap = FALSE, row_overlap = FALSE, pceru = 0.8, pcerv = 0.8, merr = 0.0001,
              iters = 100, assign_unclustered_samples = FALSE)

# Generate Table
summary_table(clinical_data, res4, Age, Sex, crp, ddimer)
```

Index