Package 'rSRD'

February 8, 2023

Type Package

Title Sum of Ranking Differences Statistical Test

Version 0.1.7

Maintainer Jochen Staudacher <jochen.staudacher@hs-kempten.de>

Description We provide an implementation for Sum of Ranking Differences (SRD), a novel statistical test introduced by Héberger (2010) <doi:10.1016/j.trac.2009.09.009>. The test allows the comparison of different solutions through a reference by first performing a rank transformation on the input, then calculating and comparing the distances between the solutions and the reference - the latter is measured in the L1 norm. The reference can be an external benchmark (e.g. an established gold standard) or can be aggregated from the data. The calculated distances, called SRD scores, are validated in two ways, see Héberger and Kollár-Hunek (2011) <doi:10.1002/cem.1320>. A randomization test (also called permutation test) compares the SRD scores of the solutions to the SRD scores of randomly generated rankings. The second validation option is cross-validation that checks whether the rankings generated from the solutions come from the same distribution or not. For a detailed analysis about the cross-validation process see Sziklai, Baranyi and Héberger (2021) <arXiv:2105.11939>. The package offers a wide array of features related to SRD including the computation of the SRD scores, validation options, input preprocessing and plotting tools.

License GPL-3

Encoding UTF-8

LinkingTo Rcpp

Imports Rcpp, dplyr, janitor, tibble, ggplot2, stringr, methods, rlang, ggrepel, gplots

SystemRequirements C++17, Rtools (>= 4.2) for Windows

NeedsCompilation yes

RoxygenNote 7.2.0

Author Jochen Staudacher [aut, cph, cre], Balázs R. Sziklai [aut, cph], Linus Olsson [aut, cph],

Dennis Horn [ctb],
Alexander Pothmann [ctb],
Ali Tugay Sen [ctb],
Attila Gere [ctb],
Károly Hébeger [ctb]

Repository CRAN

Date/Publication 2023-02-08 08:02:31 UTC

R topics documented:

`calculateCrossValidation`
calculateCrossValidation

Description

R interface to test whether the rankings induced by the columns come from the same distribution. If the number of folds and the test method are not specified, the default is the 8-fold Wilcoxon test combined with cross-validation. If the number of rows is less than 8, leave-one-out cross-validation is applied. Columns are ordered based on the SRD values of the different folds, then each consecutive column-pairs are tested. Test statistics for Alpaydin test follows F distribution with df1=2k, df2=k degrees of freedom. Dietterich test statistics follow t-distribution with k degrees of freedom (two-tailed). Wilcoxon test statistics is calculated as the absolute value of the difference of the sum of the positive ranks (W+) and sum of the negative ranks (W-). The distribution for this test statistics can be derived from the Wilcoxon signed rank distribution. For more information about the cross-validation process see Sziklai, Baranyi and Héberger (2021).

Usage

```
calculateCrossValidation(
  data_matrix,
  method = "Wilcoxon",
  number_of_folds = 8,
  precision = 5,
  output_to_file = TRUE
)
```

Arguments

data_matrix	A DataFrame.
method	A string specifying the method. The methods "Wilcoxon", "Alpaydin" and "Dietterich" are available.
number_of_folds	The number of folds used in the cross validation. Ranges between 5 to 10.
precision	The precision used for the the ranking matrix transformation.
output_to_file	Boolean flag to enable file output.

Value

A List containing

- a new column order sorted by the median of the SRD values computed on the different folds
- a vector of test statistics corresponding to each consecutive column pairs
- a vector indicating the test statistics' statistical significance
- the SRD values of different folds and
- additional data needed for the plotCrossValidation function.

Author(s)

Balázs R. Sziklai <sziklai.balazs@krtk.hu>, Linus Olsson <linusmeol@gmail.com>, Jochen Staudacher <jochen.staudacher@hs-kempten.de>

References

Sziklai, Balázs R., Máté Baranyi, and Károly Héberger (2021). "Testing Cross-Validation Variants in Ranking Environments", arXiv preprint arXiv:2105.11939 (2021).

Examples

```
df <- data.frame(
Sol_1=c(7, 6, 5, 4, 3, 2, 1),
Sol_2=c(1, 2, 3, 4, 5, 7, 6),
Sol_3=c(1, 2, 3, 4, 7, 5, 6),
Ref=c(1, 2, 3, 4, 5, 6, 7))

calculateCrossValidation(df, output_to_file = FALSE)
```

calculateSRDDistribution *calculateSRDDistribution*

Description

R interface to calculate the SRD distribution that corresponds to the data.

Usage

```
calculateSRDDistribution(
  data_matrix,
  option = "f",
  tie_probability = 0,
  output_to_file = FALSE
)
```

Arguments

`data_matrix`	A DataFrame.
`option`	A char to specify how ties are generated in the simulation. The following options are available: • 'n': There are no ties for the solution vectors, the reference vector is fixed. • 'r': There are no ties. Both the column vector and the reference are generated randomly. • 't': Ties occur with a fixed probability specified by the user for both the solution vectors and the reference vector. • 'p': Ties occur with a fixed probability specified by the user for the solution vectors, the reference vector is fixed. • 'd': Tie distribution reflects the tie frequencies displayed by the solution vectors, the reference vector is fixed. • As default (recommended): Tie distribution reflects the tie frequencies displayed in the reference, the reference vector is fixed.
`tie_probability`	The probability with which ties can occur.
`output_to_file`	Boolean flag to enable file output.

Value

A List containing the SRD distribution and related descriptive statistics. xx1 value indicates the 5 percent significance threshold. SRD values falling between xx1 and xx19 are not distinguishable from SRD scores of random rankings, while an SRD score higher than xx19 indicates that the solution ranks the objects in a reverse order (with 5 percent significance).

Author(s)

Balázs R. Sziklai <sziklai.balazs@krtk.hu>, Linus Olsson <linusmeol@gmail.com>

Examples

```
df <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))

calculateSRDDistribution(df, option = 'p', tie_probability = 0.5)
```

---

calculateSRDValues *calculateSRDValues*

Description

R interface to calculate SRD values. To test the results' significance run calculateSRDDistribution(). For more information about SRD scores and their validation see Héberger and Kollár-Hunek (2011).

Usage

```
calculateSRDValues(data_matrix, output_to_file = FALSE)
```

Arguments

data_matrix	A DataFrame.
output_to_file	Boolean flag to enable file output.

Value

A vector containing the SRD values.

Author(s)

Balázs R. Sziklai <sziklai.balazs@krtk.hu>, Linus Olsson <linusmeol@gmail.com>

References

Héberger K., Kollár-Hunek K. (2011) "Sum of ranking differences for method discrimination and its validation: comparison of ranks with random numbers", Journal of Chemometrics, 25(4), pp. 151–158.

Examples

```
df <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))

calculateSRDValues(df)
```

`plotCrossValidation` *plotCrossValidation*

Description

Plots data generated by the calculateCrossValidation function as a boxplot. Includes max and min as whiskers as well as the average (marked by a crossed circle), median (marked by a horizontal bold line) and the 1st and 3rd quartile of the values. Visualizes outliers in the data as red triangles.

Usage

```
plotCrossValidation(cv_results)
```

Arguments

`cv_results`	The List of results returned by the calculateCrossValidation function.

Value

None.

Author(s)

Linus Olsson <linusmeol@gmail.com>, Alexander Pothmann

Examples

```
df <- data.frame(
Sol_1=c(7, 6, 5, 4, 3, 2, 1),
Sol_2=c(1, 2, 3, 4, 5, 7, 6),
Sol_3=c(1, 2, 3, 4, 7, 5, 6),
Ref=c(1, 2, 3, 4, 5, 6, 7))

cv_results <- rSRD::calculateCrossValidation(df, output_to_file = FALSE)
rSRD::plotCrossValidation(cv_results)
```

plotHeatmapSRD *plotHeatmapSRD*

Description

Heatmap is generated based on the pairwise distance - measured in SRD - of the columns. Each column is set as reference once, then SRD values are calculated for the other columns.

Usage

```
plotHeatmapSRD(df, output_to_file = FALSE)
```

Arguments

df	A DataFrame.
output_to_file	Logical. If true, the distance matrix will be saved to the hard drive.

Value

Returns a heatmap and the corresponding distance matrix.

Author(s)

Attila Gere <gereattilaphd@gmail.com>, Linus Olsson <linusmeol@gmail.com>, Jochen Staudacher <jochen.staudacher@hs-kempten.de>

Examples

```
srdInput <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))

plotHeatmapSRD(srdInput)
```

plotPermTest *plotPermTest*

Description

Plots the permutation test for the given data frame by using the simulation data created by the calculateSRDDistribution() function.

Usage

```
plotPermTest(df, simulationData, densityToDistr = FALSE)
```

Arguments

df	A DataFrame.
simulationData	The output of the calculateSRDDistribution() function.
densityToDistr	Flag to display the cumulative distribution function instead of the probability density.

Value

None.

Author(s)

Linus Olsson <linusmeol@gmail.com>

Examples

```
df <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))

simulationData <- rSRD::calculateSRDDistribution(df)
plotPermTest(df, simulationData)
```

---

utilsCalculateDistance

utilsCalculateDistance

Description

Calculates the Manhattan-distance between two rankings and inserts it into the DataFrame after the first column.

Usage

```
utilsCalculateDistance(df, nameCol, refCol)
```

Arguments

df	A DataFrame.
nameCol	The current Column of the iteration.
refCol	The reference Column of the dataFrame.

Value

Returns a new df that has a Distance Column based on the nameCol.

Author(s)

Ali Tugay Sen

Examples

```
SRDInput <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))
nameCol <- "A"
refCol <- "B"
 for(i in names(SRDInput)){ SRDInput <- rSRD::utilsCalculateRank(SRDInput,i)}
rSRD::utilsCalculateDistance(SRDInput,nameCol,refCol)
```

---

utilsCalculateRank *utilsCalculateRank*

Description

Calculates the ranking of a given column.

Usage

```
utilsCalculateRank(df, nameCol)
```

Arguments

df	A DataFrame.
nameCol	The name of the column to be ranked. Note that this parameter needs to be specified as there is no default value.

Value

Returns a new df that has an additional column with the rankings of the column specified by nameCol.

Author(s)

Jochen Staudacher <jochen.staudacher@hs-kempten.de>

Examples

```
SRDInput <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))
columnName <- "A"
rSRD::utilsCalculateRank(SRDInput,columnName)
```

---

utilsColorPalette *utilsColorPalette*

Description

Unique color palette for heatmaps.

Usage

```
utilsColorPalette
```

Format

An object of class `character` of length 250.

Author(s)

Attila Gere <gereattilaphd@uni-mate.hu>, Balázs R. Sziklai <sziklai.balazs@krtk.hu>,

Examples

```
barplot(rep(1,250), col = utilsColorPalette)
```

utilsCreateReference *utilsCreateReference*

Description

Adds a new reference column based on the input DataFrame df and the given method. This function iterates over the rows and applies the given method to define the value of the reference. Available options are: max, min, median, mean and mixed. This column is appended to the DataFrame. When "mixed" is specified the function will consider the refVector for creating the reference column.

Usage

```
utilsCreateReference(df, method = "max", refVector = c())
```

Arguments

df	A DataFrame.
method	A string value specifying the reference creating method. Available options: max, min, median, mean and mixed.
refVector	A vector of strings that specifies a method for each row. Vector size should be equal to the number of rows in the DataFrame df.

Value

Returns a new DataFrame appended with the reference column created by the method.

Author(s)

Ali Tugay Sen, Linus Olsson <linusmeol@gmail.com>

Examples

```
SRDInput <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))
proc_data <- rSRD::utilsPreprocessDF(SRDInput)
ref <- c("min","max","min","max","mean")
rSRD::utilsCreateReference(proc_data, method = "mixed", ref)
```

utilsDetailedSRD *utilsDetailedSRD*

Description

Detailed calculation of the SRD values including the computation of the ranking transformation. Unless there is a column specified with referenceCol the last column will always taken as the reference.

Usage

```
utilsDetailedSRD(
  df,
  referenceCol,
  createRefCol = function() {
 }
)
```

Arguments

df	A DataFrame.
referenceCol	Optional. A string that contains a column of df which will be used as the reference column.
createRefCol	Optional. Can be max, min, median, mean. Creates a new Column based on the existing df and attaches it to df as the reference Column.

Value

Returns a new DataFrame that shows the detailed SRD computation (ranking transformation and distance calculation). A newly added row contains the SRD values (displayed without normalization).

Author(s)

Ali Tugay Sen

Examples

```
SRDInput <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))
rSRD::utilsDetailedSRD(SRDInput)
```

utilsDetailedSRDNoChars *utilsDetailedSRDNoChars*

Description

Detailed calculation of the SRD values including the computation of the ranking transformation. Unless there is a column specified with referenceCol the last column will always taken as the reference. In this variant unused variables will not be converted to chars.

Usage

```
utilsDetailedSRDNoChars(
  df,
  referenceCol,
  createRefCol = function() {
 }
)
```

Arguments

df	A DataFrame.
referenceCol	Optional. A string that contains a column of df which will be used as the reference column.
createRefCol	Optional. Can be max, min, median, mean. Creates a new Column based on the existing df and attaches it to df as the reference Column.

Value

Returns a new DataFrame that shows the detailed SRD computation (ranking transformation and distance calculation). A newly added row contains the SRD values (displayed without normalization).

Author(s)

Ali Tugay Sen

Examples

```
SRDInput <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))
rSRD::utilsDetailedSRDNoChars(SRDInput)
```

`utilsMaxSRD` *utilsMaxSRD*

Description

Calculates the maximum distance between two rankings of size n. This function is used to normalize SRD values.

Usage

```
utilsMaxSRD(rowsCount)
```

Arguments

`rowsCount`	The number of rows in the SRD calculation.

Value

The maximum achievable SRD value.

Author(s)

Dennis Horn

Examples

```
maxSRD <- rSRD::utilsMaxSRD(5)
```

`utilsPreprocessDF` *utilsPreprocessDF*

Description

This function preprocesses the DataFrame depending on the `method`.

Usage

```
utilsPreprocessDF(df, method = "range_scale")
```

Arguments

`df`	A DataFrame.
`method`	A string that should contain "scale_to_unit", "standardize", "range_scale" or "scale_to_max".

Value

Returns a new df that has a Distance Column based on the nameCol.

Author(s)

Ali Tugay Sen, Dennis Horn <dennishorn@hotmail.de>, Linus Olsson <linusmeol@gmail.com>

Examples

```
SRDInput <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))
method <- "standardize"
utilsPreprocessDF(SRDInput,method)
```

---

utilsRankingMatrix *utilsRankingMatrix*

Description

R interface to perform the rank transformation on the columns of the input data frame. Ties are resolved by fractional ranking.

Usage

```
utilsRankingMatrix(data_matrix)
```

Arguments

data_matrix	A DataFrame.

Value

A DataFrame containing the ranking matrix.

Author(s)

Balázs R. Sziklai <sziklai.balazs@krtk.hu>, Linus Olsson <linusmeol@gmail.com>

Examples

```
df <- data.frame(
A=c(32, 52, 44, 44, 47),
B=c(73, 75, 65, 76, 70),
C=c(60, 59, 57, 55, 60),
D=c(35, 24, 44, 83, 47),
E=c(41, 52, 46, 50, 65))

utilsRankingMatrix(df)
```

`utilsTieProbability` *utilsTieProbability*

Description

Calculates the tie probability for a given vector. The tie probability is defined as the number of consecutive tied component-pairs *in the sorted vector* divided by the size of the vector minus 1.

Usage

```
utilsTieProbability(x)
```

Arguments

x	A vector.

Value

Returns the tie probability as a numeric value.

Author(s)

Ali Tugay Sen, Linus Olsson <linusmeol@gmail.com>

Examples

```
x <-c(1,2,4,4,5,5,6)
rSRD::utilsTieProbability(x)
```

Index