Package 'DSAM'

January 20, 2025

Title Data Splitting Algorithms for Model Developments

Version 1.0.2

Description Providing six different algorithms that can be used to split the available data into training, test and validation subsets with similar distribution for hydrological model developments. The dataSplit() function will help you divide the data according to specific requirements, and you can refer to the par.default() function to set the parameters for data splitting. The getAUC() function will help you measure the similarity of distribution features between the data subsets. For more information about the data splitting algorithms, please refer to:
Chen et al. (2022) <doi:10.1016/j.jhydrol.2022.128340>,
Zheng et al. (2022) <doi:10.1029/2021WR031818>.

License MIT + file LICENSE

URL `https://github.com/lark-max/DSAM`

BugReports `https://github.com/lark-max/DSAM/issues`

Encoding UTF-8

RoxygenNote 7.2.3

Imports caret, kohonen, Matrix, pROC, stats, utils, xgboost

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author Feifei Zheng [aut, ths],
Junyi Chen [aut, cre] (<`https://orcid.org/0009-0001-1978-6475`>)

Maintainer Junyi Chen <`jun1chen@zju.edu.cn`>

Repository CRAN

Date/Publication 2024-01-29 14:20:03 UTC

Contents

`checkFull` *Check whether the sample set is full*

Description

Built-in function: This function includes four arguments, where the first one contains the information of the original dataset as well as the three subsets, and the remaining three augments are the maximum sample sizes for the training, test and validation subsets respectively.

Usage

```
checkFull(split.info, num.train, num.test, num.valid)
```

Arguments

`split.info`	List type, which contains the original data set, three sampling subsets, termination signal and other relevant sampling information.
`num.train`	The number of training data points specified by the user.
`num.test`	The number of test data points specified by the user.
`num.valid`	The number of validation data points specified by the user.

Value

A list with sampling information.

`dataSplit` *Main function of data splitting algorithm*

Description

'DSAM' interface function: The user needs to provide a parameter list before data-splitting. These parameters have default values, with details given in the `par.default` function. Conditioned on the parameter list, this function carries out the data-splitting based on the algorithm specified by the user. The available algorithms include the traditional time-consecutive method (TIMECON), DUPLEX, MDUPLEX SOMPLEX, SBSS.P, SS. The algorithm details can be found in Chen et al. (2022). Note that this package focuses on deals with the dataset with multiple inputs but one output, where this output is used to enable the application of various data-splitting algorithms.

Usage

```
dataSplit(data, control = list(), ...)
```

Arguments

`data` The dataset should be matrix or Data.frame. The format should be as follows: Column one is a subscript vector used to mark each data point (each row is considered as a data point); Columns from 2 to N-1 are the input data, and Column N are the output data.

`control` User-defined parameter list, where each parameter definition refers to the `par.default` function.

`...` A redundant argument list.

Value

Return the training, test and validation subsets. If the original data are required to be split into two subsets, the training and test subsets can be combined into a single calibration subset.

Author(s)

Feifei Zheng <`feifeizheng@zju.edu.cn`>

Junyi Chen <`jun1chen@zju.edu.cn`>

References

Chen, J., Zheng F., May R., Guo D., Gupta H., and Maier H. R.(2022).Improved data splitting methods for data-driven hydrological model development based on a large number of catchment samples, Journal of Hydrology, 613.

Zheng, F., Chen J., Maier H. R., and Gupta H.(2022). Achieving Robust and Transferable Performance for Conservation-Based Models of Dynamical Physical Systems, Water Resources Research, 58(5).

Zheng, F., Chen, J., Ma, Y., Chen Q., Maier H. R., and Gupta H.(2023). A Robust Strategy to Account for Data Sampling Variability in the Development of Hydrological Models, Water Resources Research, 59(3).

Examples

```
data("DSAM_test_smallData")
res.sml = dataSplit(DSAM_test_smallData)

data("DSAM_test_modData")
res.mod = dataSplit(DSAM_test_modData, list(sel.alg = "SBSS.P"))

data("DSAM_test_largeData")
res.lag = dataSplit(DSAM_test_largeData, list(sel.alg = "SOMPLEX"))
```

`DP.initialSample` *Initial sampling of DUPLEX*

Description

Built-in function: The initial sampling function of DUPLEX algorithm, aimed to obtain the two data points with the farthest Euclidean distance from the original data set and assign them to the corresponding sampling subset.

Usage

```
DP.initialSample(split.info, choice)
```

Arguments

`split.info`	A list containing relevant sampling information such as the original dataset and three sample subsets.
`choice`	The variable must be one name of the three sample subsets contained in split.info, according to which the function assigns the current two data points to the specific sampling subset.

Value

Return the training, test and validation subsets. If the original data are required to be split into two subsets, the training and test subsets can be combined into a single calibration subset.

DP.reSample *Repeat sampling of DUPLEX*

Description

Built-in function: The cyclic sampling function of DUPLEX algorithm that takes the two data points farthest from the current sampling set and assigns them to the corresponding sampling subset.

Usage

```
DP.reSample(split.info, choice)
```

Arguments

split.info	A list containing relevant sampling information such as the original dataset and three sample subsets.
choice	The variable must be one name of the three sample subsets contained in split.info, according to which the function assigns the current two data points to the specific sampling subset.

Value

Return the training, test and validation subsets. If the original data are required to be split into two subsets, the training and test subsets can be combined into a single calibration subset.

DSAM_test_largeData *large test dataset*

Description

A large dataset containing the rainfall and runoff time series using for testing data splitting algorithms

Usage

```
DSAM_test_largeData
```

Format

A data frame with 3650 rows and 5 variables

Idex Data subscript that marks the position of each data point

I input vectors

I.1 input vectors

I.2 input vectors

O The output vector, usually the runoff ...

DSAM_test_modData *Moderate test dataset*

Description

A moderate dataset containing the rainfall and runoff time series using for testing data splitting algorithms

Usage

```
DSAM_test_modData
```

Format

A data frame with 1000 rows and 5 variables

Idex Data subscript that marks the position of each data point

I input vectors

I.1 input vectors

I.2 input vectors

O The output vector, usually the runoff ...

DSAM_test_smallData *Small test dataset*

Description

A small dataset containing the rainfall and runoff time series using for testing data splitting algorithms

Usage

```
DSAM_test_smallData
```

Format

A data frame with 200 rows and 5 variables

Idex Data subscript that marks the position of each data point

I input vectors

I.1 input vectors

I.2 input vectors

O The output vector, usually the runoff ...

DUPLEX *'DSAM' - DUPLEX algorithm*

Description

The deterministic DUPLEX algorithm, with details given in Chen et al. (2022).

Usage

```
DUPLEX(data, control)
```

Arguments

`data` The dataset should be matrix or Data.frame. The format should be as follows: Column one is a subscript vector used to mark each data point (each row is considered as a data point); Columns from 2 to N-1 are the input data, and Column N are the output data.

`control` User-defined parameter list, where each parameter definition refers to the `par.default` function.

Value

Return the training, test and validation subsets. If the original data are required to be split into two subsets, the training and test subsets can be combined into a single calibration subset.

getAUC *Get the AUC value between two datasets*

Description

This function calls `[kohonen]{xgboost}` to train the classifier, followed by calculating the similarity between the two given datasets. The return value is a AUC index, ranging between 0 and 1, where the AUC is closer to 0.5, the more similar the two data sets is.

Usage

```
getAUC(data1, data2)
```

Arguments

`data1` Dataset 1, the data type must be numeric, matrix or Data.frame.

`data2` Dataset 2, the data type must be numeric, matrix or Data.frame.

Value

Return the AUC value.

getMax *Get the maximum of the output column from the original data set*

Description

This function return the maximum of runoff(output columu) for users.

Usage

```
getMax(data)
```

Arguments

`data`	The original data set, the data type must be numeric, matrix or Data.frame.

Value

Return the maximum value of the output column.

getMean *Get the mean and standard deviation of the output column from the original data set*

Description

This function return the mean and standard deviation of runoff(output columu) for users.

Usage

```
getMean(data)
```

Arguments

`data`	The original data set, the data type must be numeric, matrix or Data.frame.

Value

Return a list with mean value and standard deviation.

getMin *Get the minimum of the output column from the original data set*

Description

This function return the minimum of runoff(output columu) for users.

Usage

```
getMin(data)
```

Arguments

`data`	The original data set, the data type must be numeric, matrix or Data.frame.

Value

Return the minimum value of the output column.

getSnen *Get sampling number of each SOM neuron*

Description

Built-in function: Calculates the maximum number of samples of each subset in each neuron within the SOM network based on the sampling ratio specified by the user.

Usage

```
getSnen(som.info, control)
```

Arguments

`som.info`	The list contains information about the SOM network, including the total number of neurons, the number of rows, and the set of data points within each neuron.
`control`	User-defined parameter list, where each parameter definition refers to the par.default function.

Value

This function return a list containing three vectors Tr,Ts and Vd, the length of which is the same as the number of neurons. Tr,Ts and Vd vectors record the specified amount of data that need be obtained for the Training, Test and Validation subset in each neuron respectively.

MDUPLEX *'DSAM' - MDUPLEX algorithm*

Description

This is a modified MDUPLEX algorithm, which is also deterministic, with details given in Zheng et al. (2022).

Usage

```
MDUPLEX(data, control)
```

Arguments

`data` The dataset should be matrix or Data.frame. The format should be as follows: Column one is a subscript vector used to mark each data point (each row is considered as a data point); Columns from 2 to N-1 are the input data, and Column N are the output data.

`control` User-defined parameter list, where each parameter definition refers to the par.default function.

Value

Return the training, test and validation subsets. If the original data are required to be split into two subsets, the training and test subsets can be combined into a single calibration subset.

References

Chen, J., Zheng F., May R., Guo D., Gupta H., and Maier H. R.(2022), Improved data splitting methods for data-driven hydrological model development based on a large number of catchment samples, Journal of Hydrology, 613.

Zheng, F., Chen J., MaierH. R., and Gupta H.(2022), Achieving Robust and Transferable Performance for Conservation-Based Models of Dynamical Physical Systems, Water Resources Research, 58(5).

par.default *Default parameter list*

Description

The list of parameters needs to be set by the user, each with a default value.

include.inp Boolean variable that determines whether the input vectors should be included during the Euclidean distance calculation. The default is `TRUE`.

seed Random number seed. The default is `1000`.

sel.alg A string variable that represents the available data splitting algorithms including `"SOMPLEX"`, `"MDUPLEX"`, `"DUPLEX"`, `"SBSS.P"`, `"SS"` and `"TIMECON"`. The default is `"MDUPLEX"`.

prop.Tr The proportion of data allocated to the training subset, where the default is `0.6`.

prop.Ts The proportion of data allocated to the test subset, where the default is `0.2`.

Train A string variable representing the output file name for the training data subset. The default is `"Train.txt"`.

Test A string variable representing the output file name for the test data subset. The default is `"Test.txt"`.

Validation A string variable representing the output file name for the validation data subset. The default is `"Valid.txt"`.

loc.calib Vector type: When sel.alg = "TIMECON", the program will select a continuous time-series data subset from the original data set, where the start and end positions are determined by this vector, with the first and the second value representing the start and end position in percentage of the original dataset. The default is `c(0,0.6)`, implying that the algorithm selects the first 60% of the data from the original dataset.

writeFile Boolean variable that determines whether the data subsets need to be output or not. The default is `FALSE`.

showTrace Boolean variable that determines the level of user feedback. The default is `FALSE`.

Usage

```
par.default()
```

Value

None

`remainUnsample` *Get the remain unsampled data after* `SSsample`

Description

Built-in function: This function is used in the semi-deterministic SS algorithm, and it contains two parameters X and Y, both of which are in an increased order. All data points in X vector that have not appeared in Y vector will be recorded and returned by this function.

Usage

```
remainUnsample(X, Y)
```

Arguments

`X`	A vector that needs to be sampled.
`Y`	A vector with data samples from X.

Value

A vector containing the remaining data that are not in Y.

`SBSS.P` *'DSAM' - SBSS.P algorithm*

Description

SBSS.P algorithm is a stochastic algorithm. It obtains data subsets through uniform sampling in each neuron after clustering through SOM neural network, with details given in May et al. (2010).

Usage

```
SBSS.P(data, control)
```

Arguments

`data`	The dataset should be matrix or Data.frame. The format should be as follows: Column one is a subscript vector used to mark each data point (each row is considered as a data point); Columns from 2 to N-1 are the input data, and Column N are the output data.
`control`	User-defined parameter list, where each parameter definition refers to the par.default function.

Value

Return the training, test and validation subsets. If the original data are required to be split into two subsets, the training and test subsets can be combined into a single calibration subset.

References

May, R. J., Maier H. R., and Dandy G. C.(2010), Data splitting for artificial neural networks using SOM-based stratified sampling, Neural Netw, 23(2), 283-294.

selectData *Select specific split data*

Description

Built-in function: This function decides whether to process the input dataset according to the parameter `include.inp`. If TRUE, this function removes Column 1 of the input dataset; otherwise, it returns the Column N of the data set.

Usage

```
selectData(data, control)
```

Arguments

`data`	The dataset should be matrix or Data.frame. The format should be as follows: Column one is a subscript vector used to mark each data point (each row is considered as a data point); Columns from 2 to N-1 are the input data, and Column N are the output data.
`control`	User-defined parameter list, where each parameter definition refers to the par.default function.

Value

Returns a matrix for subsequent calculations.

somCluster *Self-organized map clustering*

Description

Built-in function: This function performs clustering for a given dataset by calling the `[kohonen]{som}` function from a "kohonen" package.

Usage

```
somCluster(data)
```

Arguments

`data`	The dataset in matrix or data.frame, containing only input and output vectors, but with no subscript vector.

Value

Return a data list of clustering neurons in the SOM network.

SOMPLEX *'DSAM' - SOMPLEX algorithm*

Description

SOMPLEX algorithm is a stochastic algorithm, with details given in Chen et al. (2022) and Zheng et al. (2023)

Usage

```
SOMPLEX(data, control)
```

Arguments

`data`	The dataset should be matrix or Data.frame. The format should be as follows: Column one is a subscript vector used to mark each data point (each row is considered as a data point); Columns from 2 to N-1 are the input data, and Column N are the output data.
`control`	User-defined parameter list, where each parameter definition refers to the `par.default` function.

Value

Return the training, test and validation subsets. If the original data are required to be split into two subsets, the training and test subsets can be combined into a single calibration subset.

References

Chen, J., Zheng F., May R., Guo D., Gupta H., and Maier H. R.(2022), Improved data splitting methods for data-driven hydrological model development based on a large number of catchment samples, Journal of Hydrology, 613.

SS *'DSAM' - SS algorithm*

Description

The systematic stratified (SS) is a semi-deterministic method, with details given in Zheng et al. (2018).

Usage

```
SS(data, control)
```

Arguments

`data` The type of data set to be divided should be matrix or Data.frame, and the data format is as follows: The first column is a subscript vector, which is used to mark each data point (each row is regarded as a data point); Columns 2 through N-1 are the input vectors, and columns N (the last) are the output vectors.

`control` User-defined parameter list, where each parameter definition refers to the `par.default` function.

Value

Return the training, test and validation subsets. If the original data are required to be split into two subsets, the training and test subsets can be combined into a single calibration subset.

References

Zheng, F., Maier, H.R., Wu, W., Dandy, G.C., Gupta, H.V. and Zhang, T. (2018) On Lack of Robustness in Hydrological Model Development Due to Absence of Guidelines for Selecting Calibration and Evaluation Data: Demonstration for Data-Driven Models. Water Resources Research 54(2), 1013-1030.

`SSsample` *Core function of SS sampling*

Description

Built-in function: This function performs the SS algorithm.

Usage

```
SSsample(index, prop)
```

Arguments

`index` A subscript vector whose subscript corresponds to the output vector of the data point sorted in an ascending order.

`prop` The sampling ratio, with the value ranging between 0 and 1.

Value

Return a vector containing the subscript of the sampled data points.

`standardise` *Standardized data*

Description

Built-in function: This function is used to standardize the data.

Usage

```
standardise(data)
```

Arguments

`data`	The dataset should be of type matrix or Data.frame and contain only the input and output vectors.

Value

Return a matrix with normalized data.

`TIMECON` *'DSAM' - Time-consecutive algorithm*

Description

This function selects a time-consecutive data from the original data set as the calibration (training and test) subset, and the remaining data is taken as the evaluation subset.

Usage

```
TIMECON(data, control)
```

Arguments

`data`	The dataset should be matrix or Data.frame. The format should be as follows: Column one is a subscript vector used to mark each data point (each row is considered as a data point); Columns from 2 to N-1 are the input data, and Column N are the output data.
`control`	User-defined parameter list, where each parameter definition refers to the `par.default` function.

Value

Return the calibration and validation subsets.

Index