

Package ‘SoftClustering’

July 21, 2025

Type Package

Title Soft Clustering Algorithms

Description It contains soft clustering algorithms, in particular approaches derived from rough set theory: Lingras & West original rough k-means, Peters' refined rough k-means, and PI rough k-means. It also contains classic k-means and a corresponding illustrative demo.

Version 2.1.3

Author G. Peters (Ed.)

Maintainer G. Peters <peters.activities@gmail.com>

Depends R (>= 4.1)

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-18 07:52:35 UTC

Contents

createLowerMShipMatrix	2
datatypeInteger	2
DemoDataC2D2a	3
HardKMeans	3
HardKMeansDemo	4
initializeMeansMatrix	5
initMeansC2D2a	6
initMeansC3D2a	7
initMeansC4D2a	7
initMeansC5D2a	8
normalizeMatrix	8
plotRoughKMeans	9

RoughKMeans_LW	10
RoughKMeans_PE	11
RoughKMeans_PI	12
RoughKMeans_SHELL	14

Index	16
--------------	-----------

<code>createLowerMShipMatrix</code>	<i>Create Lower Approximation</i>
-------------------------------------	-----------------------------------

Description

Creates a lower approximation out of an upper approximation.

Usage

`createLowerMShipMatrix(upperMShipMatrix)`

Arguments

`upperMShipMatrix`
An upper approximation matrix.

Value

Returns the corresponding lower approximation.

Author(s)

G. Peters.

<code>datatypeInteger</code>	<i>Rough k-Means Plotting</i>
------------------------------	-------------------------------

Description

Checks for integer.

Usage

`datatypeInteger(x)`

Arguments

`x` As a replacement for `is.integer()`. `is.integer()` delivers FALSE when the variable is numeric (as superset for integer etc.)

Value

TRUE if x is integer otherwise FALSE.

Author(s)

G. Peters.

DemoDataC2D2a	<i>A small two-dimensional dataset with two clusters for demonstration purposes. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().</i>
---------------	---

Description

A small two-dimensional dataset with two clusters for demonstration purposes. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().

Usage

```
data(DemoDataC2D2a)
```

Format

Rows: objects, columns: features

Examples

```
data(DemoDataC2D2a)
```

HardKMeans	<i>Hard k-Means</i>
------------	---------------------

Description

HardKMeans performs classic (hard) k-means.

Usage

```
HardKMeans(dataMatrix, meansMatrix, nClusters, maxIterations)
```

Arguments

<code>dataMatrix</code>	Matrix with the objects to be clustered. Dimension: [nObjects x nFeatures].
<code>meansMatrix</code>	Select means derived from 1 = random (unity interval), 2 = maximum distances, matrix [nClusters x nFeatures] = self-defined means. Default: 2 = maximum distances.
<code>nClusters</code>	Number of clusters: Integer in [2, nObjects). Note, nCluster must be set even when meansMatrix is a matrix. For transparency, nClusters will not be overridden by the number of clusters derived from meansMatrix. Default: nClusters=2.
<code>maxIterations</code>	Maximum number of iterations. Default: maxIterations=100.

Value

`$upperApprox`: Obtained upper approximations [nObjects x nClusters]. Note: Apply function `createLowerMShipMatrix()` to obtain lower approximations; and for the boundary: `boundary = upperApprox - lowerApprox`.

`$clusterMeans`: Obtained means [nClusters x nFeatures].

`$nIterations`: Number of iterations.

Author(s)

M. Goetz, G. Peters, Y. Richter, D. Sacker, T. Wochinger.

References

Lloyd, S.P. (1982) Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28**, 128–137. <doi:10.1016/j.ijar.2012.10.003>.

Peters, G.; Crespo, F.; Lingras, P. and Weber, R. (2013) Soft clustering – fuzzy and rough approaches and their extensions and derivatives. *International Journal of Approximate Reasoning* **54**, 307–322. <doi:10.1016/j.ijar.2012.10.003>.

Examples

```
# An illustrative example clustering the sample data set DemoDataC2D2a.txt
HardKMeans(DemoDataC2D2a, 2, 2, 100)
```

HardKMeansDemo

Hard k-Means Demo

Description

HardKMeansDemo shows how hard k-means performs stepwise. The number of features is set to 2 and the maximum number of iterations is 100.

Usage

```
HardKMeansDemo(dataMatrix, meansMatrix, nClusters)
```

Arguments

dataMatrix	Matrix with the objects to be clustered. Dimension: [nObjects x nFeatures]. Default: no default set.
meansMatrix	Select means derived from 1 = random (unity interval), 2 = maximum distances, matrix [nClusters x nFeatures=2] = self-defined means. Default: meansMatrix=1 (random).
nClusters	Number of clusters: Integer in [2, min(5, nObjects-1)]. Note, nCluster must be set even when meansMatrix is a matrix. For transparency, nClusters will not be overridden by the number of clusters derived from meansMatrix. Default: nClusters=2.

Value

None.

Author(s)

G. Peters.

References

- Lloyd, S.P. (1982) Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28**, 128–137. <doi:10.1016/j.ijar.2012.10.003>.
- Peters, G.; Crespo, F.; Lingras, P. and Weber, R. (2013) Soft clustering – fuzzy and rough approaches and their extensions and derivatives. *International Journal of Approximate Reasoning* **54**, 307–322. <doi:10.1016/j.ijar.2012.10.003>.

Examples

```
# Clustering the data set DemoDataC2D2a.txt (nClusters=2, random initial means)
HardKMeansDemo(DemoDataC2D2a,1,2)
# Clustering the data set DemoDataC2D2a.txt (nClusters=2,3,4; initially set means)
HardKMeansDemo(DemoDataC2D2a,initMeansC2D2a,2)
HardKMeansDemo(DemoDataC2D2a,initMeansC3D2a,3)
HardKMeansDemo(DemoDataC2D2a,initMeansC4D2a,4)
# Clustering the data set DemoDataC2D2a.txt (nClusters=5, initially set means)
# It leads to an empty cluster: a (rare) case for an abnormal termination of k-means.
HardKMeansDemo(DemoDataC2D2a,initMeansC5D2a,5)
```

initializeMeansMatrix *Initialize Means Matrix*

Description

initializeMeansMatrix delivers an initial means matrix.

Usage

```
initializeMeansMatrix(dataMatrix, nClusters, meansMatrix)
```

Arguments

dataMatrix	Matrix with the objects as basis for the means matrix.
nClusters	Number of clusters.
meansMatrix	Select means derived from 1 = random (unity interval), 2 = maximum distances, matrix [nClusters x nFeatures] = self-defined means (will be returned unchanged). Default: 2 = maximum distances.

Value

Initial means matrix [nClusters x nFeatures].

Author(s)

M. Goetz, G. Peters, Y. Richter, D. Sacker, T. Wochinger.

initMeansC2D2a	<i>Two-dimensional dataset with two initial cluster means for the dataset DemoDataC2D2a. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().</i>
----------------	---

Description

Two-dimensional dataset with two initial cluster means for the dataset DemoDataC2D2a. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().

Usage

```
data(initMeansC2D2a)
```

Format

Rows: objects, columns: features

Examples

```
data(initMeansC2D2a)
```

initMeansC3D2a	<i>Two-dimensional dataset with three initial cluster means for the dataset DemoDataC2D2a. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().</i>
----------------	---

Description

Two-dimensional dataset with three initial cluster means for the dataset DemoDataC2D2a. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().

Usage

```
data(initMeansC3D2a)
```

Format

Rows: objects, columns: features

Examples

```
data(initMeansC3D2a)
```

initMeansC4D2a	<i>Two-dimensional dataset with four initial cluster means for the dataset DemoDataC2D2a. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().</i>
----------------	--

Description

Two-dimensional dataset with four initial cluster means for the dataset DemoDataC2D2a. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().

Usage

```
data(initMeansC4D2a)
```

Format

Rows: objects, columns: features

Examples

```
data(initMeansC4D2a)
```

initMeansC5D2a	<i>Two-dimensional dataset with five initial cluster means for the dataset DemoDataC2D2a. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().</i>
----------------	--

Description

Two-dimensional dataset with five initial cluster means for the dataset DemoDataC2D2a. See examples in the Help/Description of a function, e.g. for HardKMeansDemo().

Usage

```
data(initMeansC5D2a)
```

Format

Rows: objects, columns: features

Examples

```
data(initMeansC5D2a)
```

normalizeMatrix	<i>Matrix Normalization</i>
-----------------	-----------------------------

Description

normalizeMatrix delivers a normalized matrix.

Usage

```
normalizeMatrix(dataMatrix, normMethod, bycol)
```

Arguments

dataMatrix	Matrix with the objects to be normalized.
normMethod	1 = unity interval, 2 = normal distribution (sample variance), 3 = normal distribution (population variance). Any other value returns the matrix unchanged. Default: meansMatrix = 1 (unity interval).
bycol	TRUE = columns are normalized, i.e., each column is considered separately (e.g., in case of the unity interval and a column colA: max(colA)=1 and min(colA)=0). For bycol = FALSE rows are normalized. Default: bycol = TRUE (columns are normalized).

Value

Normalized matrix.

Author(s)

M. Goetz, G. Peters, Y. Richter, D. Sacker, T. Wochinger.

plotRoughKMeans	<i>Rough k-Means Plotting</i>
-----------------	-------------------------------

Description

plotRoughKMeans plots the rough clustering results in 2D. Note: Plotting is limited to a maximum of 5 clusters.

Usage

```
plotRoughKMeans(dataMatrix, upperMShipMatrix, meansMatrix, plotDimensions, colouredPlot)
```

Arguments

dataMatrix	Matrix with the objects to be plotted.
upperMShipMatrix	Corresponding matrix with upper approximations.
meansMatrix	Corresponding means matrix.
plotDimensions	An integer vector of the length 2. Defines the to be plotted feature dimensions, i.e., $\max(\text{plotDimensions} = \text{c}(1:2)) \leq \text{nFeatures}$. Default: $\text{plotDimensions} = \text{c}(1:2)$.
colouredPlot	Select TRUE = colouredPlot plot, FALSE = black/white plot.

Value

2D-plot of clustering results. The boundary objects are represented by stars (*).

Author(s)

G. Peters.

RoughKMeans_LW

*Lingras & West's Rough k-Means***Description**

RoughKMeans_LW performs Lingras & West's k-means clustering algorithm. The commonly accepted relative threshold is applied.

Usage

```
RoughKMeans_LW(dataMatrix, meansMatrix, nClusters, maxIterations, threshold, weightLower)
```

Arguments

dataMatrix	Matrix with the objects to be clustered. Dimension: [nObjects x nFeatures].
meansMatrix	Select means derived from 1 = random (unity interval), 2 = maximum distances, matrix [nClusters x nFeatures] = self-defined means. Default: 2 = maximum distances.
nClusters	Number of clusters: Integer in [2, nObjects). Note, nCluster must be set even when meansMatrix is a matrix. For transparency, nClusters will not be overridden by the number of clusters derived from meansMatrix. Default: nClusters=2.
maxIterations	Maximum number of iterations. Default: maxIterations=100.
threshold	Relative threshold in rough k-means algorithms (threshold >= 1.0). Default: threshold = 1.5.
weightLower	Weight of the lower approximation in rough k-means algorithms (0.0 <= weightLower <= 1.0). Default: weightLower = 0.7.

Value

\$upperApprox: Obtained upper approximations [nObjects x nClusters]. Note: Apply function createLowerMShipMatrix() to obtain lower approximations; and for the boundary: boundary = upperApprox - lowerApprox.

\$clusterMeans: Obtained means [nClusters x nFeatures].

\$nIterations: Number of iterations.

Author(s)

M. Goetz, G. Peters, Y. Richter, D. Sacker, T. Wochinger.

References

Lingras, P. and West, C. (2004) Interval Set Clustering of web users with rough k-means. *Journal of Intelligent Information Systems* **23**, 5–16. <doi:10.1023/b:jiis.0000029668.88665.1a>.

Peters, G. (2006) Some refinements of rough k-means clustering. *Pattern Recognition* **39**, 1481–1491. <doi:10.1016/j.patcog.2006.02.002>.

Lingras, P. and Peters, G. (2011) Rough Clustering. *WIREs Data Mining and Knowledge Discovery* **1**, 64–72. <doi:10.1002/widm.16>.

Lingras, P. and Peters, G. (2012) Applying rough set concepts to clustering. In: Peters, G.; Lingras, P.; Slezak, D. and Yao, Y. Y. (Eds.) *Rough Sets: Selected Methods and Applications in Management and Engineering*, Springer, 23–37. <doi:10.1007/978-1-4471-2760-4_2>.

Peters, G.; Crespo, F.; Lingras, P. and Weber, R. (2013) Soft clustering – fuzzy and rough approaches and their extensions and derivatives. *International Journal of Approximate Reasoning* **54**, 307–322. <doi:10.1016/j.ijar.2012.10.003>.

Peters, G. (2014) Rough clustering utilizing the principle of indifference. *Information Sciences* **277**, 358–374. <doi:10.1016/j.ins.2014.02.073>.

Peters, G. (2015) Is there any need for rough clustering? *Pattern Recognition Letters* **53**, 31–37. <doi:10.1016/j.patrec.2014.11.003>.

Examples

```
# An illustrative example clustering the sample data set DemoDataC2D2a.txt
RoughKMeans_LW(DemoDataC2D2a, 2, 2, 100, 1.5, 0.7)
```

RoughKMeans_PE

Peters' Rough k-Means

Description

RoughKMeans_PE performs Peters' k-means clustering algorithm.

Usage

```
RoughKMeans_PE(dataMatrix, meansMatrix, nClusters, maxIterations, threshold, weightLower)
```

Arguments

dataMatrix	Matrix with the objects to be clustered. Dimension: [nObjects x nFeatures].
meansMatrix	Select means derived from 1 = random (unity interval), 2 = maximum distances, matrix [nClusters x nFeatures] = self-defined means. Default: 2 = maximum distances.
nClusters	Number of clusters: Integer in [2, nObjects). Note, nCluster must be set even when meansMatrix is a matrix. For transparency, nClusters will not be overridden by the number of clusters derived from meansMatrix. Default: nClusters=2.
maxIterations	Maximum number of iterations. Default: maxIterations=100.
threshold	Relative threshold in rough k-means algorithms (threshold >= 1.0). Default: threshold = 1.5.
weightLower	Weight of the lower approximation in rough k-means algorithms (0.0 <= weightLower <= 1.0). Default: weightLower = 0.7.

Value

\$upperApprox: Obtained upper approximations [nObjects x nClusters]. Note: Apply function createLowerMShipMatrix() to obtain lower approximations; and for the boundary: boundary = upperApprox - lowerApprox.

\$clusterMeans: Obtained means [nClusters x nFeatures].

\$nIterations: Number of iterations.

Author(s)

M. Goetz, G. Peters, Y. Richter, D. Sacker, T. Wochinger.

References

Peters, G. (2006) Some refinements of rough k-means clustering. *Pattern Recognition* **39**, 1481–1491. <doi:10.1016/j.patcog.2006.02.002>.

Peters, G.; Crespo, F.; Lingras, P. and Weber, R. (2013) Soft clustering – fuzzy and rough approaches and their extensions and derivatives. *International Journal of Approximate Reasoning* **54**, 307–322. <doi:10.1016/j.ijar.2012.10.003>.

Peters, G. (2014) Rough clustering utilizing the principle of indifference. *Information Sciences* **277**, 358–374. <doi:10.1016/j.ins.2014.02.073>.

Peters, G. (2015) Is there any need for rough clustering? *Pattern Recognition Letters* **53**, 31–37. <doi:10.1016/j.patrec.2014.11.003>.

Examples

```
# An illustrative example clustering the sample data set DemoDataC2D2a.txt
RoughKMeans_PE(DemoDataC2D2a, 2, 2, 100, 1.5, 0.7)
```

RoughKMeans_PI

PI *Rough k-Means*

Description

RoughKMeans_PI performs pi k-means clustering algorithm in its standard case. Therefore, weights are not required.

Usage

```
RoughKMeans_PI(dataMatrix, meansMatrix, nClusters, maxIterations, threshold)
```

Arguments

<code>dataMatrix</code>	Matrix with the objects to be clustered. Dimension: [nObjects x nFeatures].
<code>meansMatrix</code>	Select means derived from 1 = random (unity interval), 2 = maximum distances, matrix [nClusters x nFeatures] = self-defined means. Default: 2 = maximum distances.
<code>nClusters</code>	Number of clusters: Integer in [2, nObjects). Note, nCluster must be set even when meansMatrix is a matrix. For transparency, nClusters will not be overridden by the number of clusters derived from meansMatrix. Default: nClusters=2.
<code>maxIterations</code>	Maximum number of iterations. Default: maxIterations=100.
<code>threshold</code>	Relative threshold in rough k-means algorithms (threshold >= 1.0). Default: threshold = 1.5.

Value

`$upperApprox`: Obtained upper approximations [nObjects x nClusters]. Note: Apply function `createLowerMShipMatrix()` to obtain lower approximations; and for the boundary: `boundary = upperApprox - lowerApprox`.

`$clusterMeans`: Obtained means [nClusters x nFeatures].

`$nIterations`: Number of iterations.

Author(s)

M. Goetz, G. Peters, Y. Richter, D. Sacker, T. Wochinger.

References

- Peters, G. (2006) Some refinements of rough k-means clustering. *Pattern Recognition* **39**, 1481–1491. <doi:10.1016/j.patcog.2006.02.002>.
- Peters, G.; Crespo, F.; Lingras, P. and Weber, R. (2013) Soft clustering – fuzzy and rough approaches and their extensions and derivatives. *International Journal of Approximate Reasoning* **54**, 307–322. <doi:10.1016/j.ijar.2012.10.003>.
- Peters, G. (2014) Rough clustering utilizing the principle of indifference. *Information Sciences* **277**, 358–374. <doi:10.1016/j.ins.2014.02.073>.
- Peters, G. (2015) Is there any need for rough clustering? *Pattern Recognition Letters* **53**, 31–37. <doi:10.1016/j.patrec.2014.11.003>.

Examples

```
# An illustrative example clustering the sample data set DemoDataC2D2a.txt
RoughKMeans_PI(DemoDataC2D2a, 2, 2, 100, 1.5)
```

RoughKMeans_SHELL	<i>Rough k-Means Shell</i>
-------------------	----------------------------

Description

RoughKMeans_SHELL performs rough k-means algorithms with options for normalization and a 2D-plot of the results.

Usage

```
RoughKMeans_SHELL(clusterAlgorithm, dataMatrix, meansMatrix, nClusters,
                    normalizationMethod, maxIterations, plotDimensions,
                    colouredPlot, threshold, weightLower)
```

Arguments

clusterAlgorithm	Select 0 = classic k-means, 1 = Lingras & West's rough k-means, 2 = Peters' rough k-means, 3 = π rough k-means. Default: clusterAlgorithm = 3 (π rough k-means).
dataMatrix	Matrix with the objects to be clustered. Dimension: [nObjects x nFeatures].
meansMatrix	Select means derived from 1 = random (unity interval), 2 = maximum distances, matrix [nClusters x nFeatures] = self-defined means. Default: 2 = maximum distances.
nClusters	Number of clusters: Integer in [2, nObjects). Note, nCluster must be set even when meansMatrix is a matrix. For transparency, nClusters will not be overridden by the number of clusters derived from meansMatrix. Default: nClusters=2. Note: Plotting is limited to a maximum of 5 clusters.
normalizationMethod	1 = unity interval, 2 = normal distribution (sample variance), 3 = normal distribution (population variance). Any other value returns the matrix unchanged. Default: meansMatrix = 1 (unity interval).
maxIterations	Maximum number of iterations. Default: maxIterations=100.
plotDimensions	An integer vector of the length 2. Defines the to be plotted feature dimensions, i.e., $\max(\text{plotDimensions} = c(1:2)) \leq n\text{Features}$. Default: plotDimensions = c(1:2).
colouredPlot	Select TRUE = colouredPlot plot, FALSE = black/white plot.
threshold	Relative threshold in rough k-means algorithms (threshold ≥ 1.0). Default: threshold = 1.5. Note: It can be ignored for classic k-means.
weightLower	Weight of the lower approximation in rough k-means algorithms ($0.0 \leq \text{weightLower} \leq 1.0$). Default: weightLower = 0.7. Note: It can be ignored for classic k-means and π rough k-means

Value

2D-plot of clustering results. The boundary objects are represented by stars (*).

\$upperApprox: Obtained upper approximations [nObjects x nClusters]. Note: Apply function createLowerMShipMatrix() to obtain lower approximations; and for the boundary: boundary = upperApprox - lowerApprox.

\$clusterMeans: Obtained means [nClusters x nFeatures].

\$nIterations: Number of iterations.

Author(s)

M. Goetz, G. Peters, Y. Richter, D. Sacker, T. Wochinger.

References

- Lloyd, S.P. (1982) Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28**, 128–137. <doi:10.1016/j.ijar.2012.10.003>.
- Lingras, P. and West, C. (2004) Interval Set Clustering of web users with rough k-means. *Journal of Intelligent Information Systems* **23**, 5–16. <doi:10.1023/b:jiis.0000029668.88665.1a>.
- Peters, G. (2006) Some refinements of rough k-means clustering. *Pattern Recognition* **39**, 1481–1491. <doi:10.1016/j.patcog.2006.02.002>.
- Lingras, P. and Peters, G. (2011) Rough Clustering. *WIREs Data Mining and Knowledge Discovery* **1**, 64–72. <doi:10.1002/widm.16>.
- Lingras, P. and Peters, G. (2012) Applying rough set concepts to clustering. In: Peters, G.; Lingras, P.; Slezak, D. and Yao, Y. Y. (Eds.) *Rough Sets: Selected Methods and Applications in Management and Engineering*, Springer, 23–37. <doi:10.1007/978-1-4471-2760-4_2>.
- Peters, G.; Crespo, F.; Lingras, P. and Weber, R. (2013) Soft clustering – fuzzy and rough approaches and their extensions and derivatives. *International Journal of Approximate Reasoning* **54**, 307–322. <doi:10.1016/j.ijar.2012.10.003>.
- Peters, G. (2014) Rough clustering utilizing the principle of indifference. *Information Sciences* **277**, 358–374. <doi:10.1016/j.ins.2014.02.073>.
- Peters, G. (2015) Is there any need for rough clustering? *Pattern Recognition Letters* **53**, 31–37. <doi:10.1016/j.patrec.2014.11.003>.

Examples

```
# An illustrative example clustering the sample data set DemoDataC2D2a.txt
RoughKMeans_SHELL(3, DemoDataC2D2a, 2, 2, 1, 100, c(1:2), TRUE, 1.5, 0.7)
```

Index

* **datasets**

- DemoDataC2D2a, [3](#)
- initMeansC2D2a, [6](#)
- initMeansC3D2a, [7](#)
- initMeansC4D2a, [7](#)
- initMeansC5D2a, [8](#)

createLowerMSHIPMatrix, [2](#)

datatypeInteger, [2](#)
DemoDataC2D2a, [3](#)

HardKMeans, [3](#)
HardKMeansDemo, [4](#)

initializeMeansMatrix, [5](#)
initMeansC2D2a, [6](#)
initMeansC3D2a, [7](#)
initMeansC4D2a, [7](#)
initMeansC5D2a, [8](#)

normalizeMatrix, [8](#)

plotRoughKMeans, [9](#)

RoughKMeans_LW, [10](#)
RoughKMeans_PE, [11](#)
RoughKMeans_PI, [12](#)
RoughKMeans_SHELL, [14](#)