Package 'clusTransition'

July 22, 2025

Type Package

Title Monitor Changes in Cluster Solutions of Dynamic Datasets

Version 1.0

Author Muhammad Atif

Maintainer Muhammad Atif <muhammad96_atif@yahoo.com>

Description Monitor and trace changes in clustering solutions of accumulating datasets at successive time points. The clusters can adopt External and Internal transition at succeeding time points. The External transitions comprise of Survived, Merged, Split, Disappeared, and newly Emerged candidates. In contrast, Internal transition includes changes in location and cohesion of the survived clusters. The package uses MONIC framework developed by
Spiliopoulou, Ntoutsi, Theodoridis, and Schult (2006)<doi:10.1145/1150402.1150491> .

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.5), flexclust, methods, graphics

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-02-22 20:20:02 UTC

Contents

`Clustering-class` *Class Clustering*

Description

Partition data into clusters

Details

Object of class `Clustering` containing clustering solution of cumulative dataset D_i. The object of class `Clustering` comprise of four slots. Slot `Clusters` contain data items of each cluster, slot `Centers` contain cluster centers, slot `k` contain the number of centers, while slot `clusterMem` contain cluster memberships vector.

Slots

`Cluster` List of matrices, where each element of the list include data items belonging to the corresponding cluster.

`Centers` Matrix of cluster centers.

`k` Number of centers.

`clusterMem` Numeric vector of cluster membership.

`Clusters` *Clustering.*

Description

Initialize slots of class `Clustering` by partitioning the dataset into `k` clusters.

Usage

```
Clusters(object, x, k)

## S4 method for signature 'Clustering,matrix,numeric'
Clusters(object, x, k)
```

Arguments

object	An object of class Clustering.
x	Numeric matrix of data.
k	Number of centers.

Details

Runs cclust function from "flexclust" package with default settings i.e. method = "kmeans", dist = "euclidean", and partition the dataset. Returns object of class Clustering.

Value

An object of class Clustering

Data2D *Synthetic Datasets (Two Dimensional)*

Description

A list of datasets generated at four time points containing two variables and cluster membership at each point.

Usage

```
Data2D
```

Format

A data frame

x1 X1.

x2 X2.

class Class membership.

Data3D *Synthetic Datasets (Three Dimensional)*

Description

A list of datasets generated at four points containing three variables and cluster membership at each point.

Usage

```
Data3D
```

Format

A data frame

x1 X1.

x2 X2.

x3 X3.

class Class membership.

extTransitionCan *External Transition Candidate.*

Description

This S4 method trace cluster solutions of dynamic dataset, and identify the candidates that experience external transition from first clustering and emerged at second clustering.

Usage

```
extTransitionCan(object)

## S4 method for signature 'TransitionCan'
extTransitionCan(object)
```

Arguments

object	An object of class Transitioncan

Value

Return an object of class TransitionCan

extTransitionCount *External Transition Count*

Description

Trace cluster solutions of dynamic datasets and count the number of clusters that experiences external transition from first clustering. The external transition includes survived, split into various daughters, spliced into one, disappeared, and newly emerged candidates.

Usage

```
extTransitionCount(object)

## S4 method for signature 'TransitionCount'
extTransitionCount(object)
```

Arguments

object	An object of class Transitioncount

Value

Return an object of class TransitionCount

internalTransition *Internal Transition Candidates.*

Description

This method identify internal transition of the survived clusters, obtained from 'extTransitionCan()' method.

Trace clustering solutions of cumulative datasets and identify the survived clusters experiencing Internal transitions. Internal transition includes the change in location and density of the survived candidates.

Usage

```
internalTransition(object)

## S4 method for signature 'intTransitionCan'
internalTransition(object)
```

Arguments

object	An object of class intTransitionCan

Value

Return an object of class `intTransitionCan`

`intTransitionCan-class` *Internal Transition Candidates*

Description

Class containing results of Internal Transition of survived clusters from first clustering ξ_1.

Arguments

`object`	An object of class Transitioncan

Slots

`Location.diff` Vector of integers containing difference in location (= Distance bw cluster centers/min(rx,ry)).

`Compactness.diff` Vector of integers containing Change in density of survived clusters (d(rx, ry)).

`Location_thrHold` Minimum value of threshold for shift in location.

`Density_thrHold` Minimum value of threshold for change in density.

`ShiftLocCan` Vector of integers containing Survived candidates with shift in their location.

`NoShiftLocCan` Vector of integers containing Survived candidates with no Shift in their Location.

`MoreCompactCan` Vector of integers containing Survived Candidates Which becomes more compact.

`MoreDiffuseCan` Vector of integers containing Survived Candidates Which becomes more diffuse.

`NoChangeCompactCan` Vector of integers containing Survived candidates with no change in compactness.

`Monic` *An S4 class that contain time steps*

Description

An S4 class that contain time steps

Arguments

`object`	An object of class `Transitioncan`

Slots

`TimeStep` Time Steps

moplot *plot Method for output*

Description

This method plot 3 barplot and 1 line graph. The first stack barplot shows SurvivalRatio and AbsorptionRatio, second barplot shows number of newly emerged clusters at each time stamp, third barplot shows number of disapeared clusters at each time stamp. The line graph shows passforward Ratio and Survival Ratio.

Usage

```
moplot(object)

## S4 method for signature 'Monic'
moplot(object)
```

Arguments

object	An object of class Monic

Overlap *Overlap*

Description

Initialize slots of class OverLap by importing clustering solutions of dynamic datasets at two consecutive time points. Clusters at each time point should be provided as a list of matrices, where each matrix contains dataset belongs to the corresponding cluster.

Usage

```
Overlap(object, e1, e2)

## S4 method for signature 'OverLap,Clustering,Clustering'
Overlap(object, e1, e2)

## S4 method for signature 'OverLap,ANY,ANY'
Overlap(object, e1, e2)
```

Arguments

object	An object of class OverLap
e1	An object of class Clustering, or any object that can be coerced, such as list of matrices or data frames that contain clusters from first clustering.
e2	An object of class Clustering, or any object that can be coerced, such as list of matrices or data frames that contain clusters from second clustering.

Value

Return an object of class `OverLap`.

`OverLap-class` *Overlap between clusters*

Description

Contains matrix of similarity indices between clusters, after clustering dynamic datasets at consecutive time points.

Slots

`Overlap` A numeric matrix containing the similarity index between clusters extracted at time point `t_1` and `t_2`. The rows of the matrix illustrate clusters extracted from first clustering $\xi_1(timepointt_1)$,whereas columns represent clusters extracted from second clustering $\xi_2(timepointt_2)$.

`rx` A numeric vector containg radius of each cluster from first clustering ξ_1.

`ry` A numeric vector containg radius of each cluster from second clustering ξ_2.

`Centersx` A numeric vector containing centers of clusters from first clustering ξ_1.

`Centersy` A numeric vector containing centers of clusters from second clustering ξ_2.

`avgDisx` A numeric vector containing average distance between points in a cluster from its center in first clustering ξ_1.

`avgDisy` A numeric vector containing average distance between points in a cluster from its center in second clustering ξ_2.

`clusterMem` A vector of integers containing cluster membership from second clustering ξ_2.

`show,Monic-method` *Show Method for output*

Description

Show Method for output

Usage

```
## S4 method for signature 'Monic'
show(object)
```

Arguments

`object`	An object of class Monic

`Transition` *Monitor Transitions in Cluster Solutions.*

Description

Model and trace the evolution of clusters evolving over time in cumulative datasets. A typical call to `Transition()` function involves three essential pieces: the data input (`listdata`, `listclus`, `overlap`), choice of window `swSize`, and the threshold parameters. The function either receive a list of datasets arriving at time points `t_1`, `t_2`, `t_3`, ..., `t_n` respectively, list of clustering solutions extracted from cumulative datasets at successive time points, or list of objects of class `OverLap` (see **Details**).

Usage

```
Transition(
  listdata,
  swSize = 1,
  Overlap = NULL,
  listclus = NULL,
  typeind = 1,
  Survival_thrHold = 0.7,
  Split_thrHold = 0.3,
  location_thrHold = 0.3,
  density_thrHold = 0.3,
  k = NULL
)
```

Arguments

`listdata`	List of numeric matrices containing datasets `d_1`, `d_2`, ..., `d_n`, or a list of objects that can be coerced to such matrices, for instance, data frames. Each element of the list contain dataset `d_i` evolving at corresponding time point `t_i`. The number of clusters in each accumulative data matrix is specified by the argument `k`.
`swSize`	Integer value (1, length(listdata)) indicating size of the sliding window. As time goes by, each window consist only objects that fall in the interval [t-swSize+1, t], while older objects are discarded. The default value of `swSize = 1` indicate landmark window model, where objects over the entire history are included i.e. [1, t]. Size of sliding window can only be provided if `listdata` arguments is choosen. If there are total `n` time stamps and a window of size `swSize` is selected then entire history would be devided into `n-swSize+2` window panes.
`Overlap`	A list of objects as produced by the `Overlap()` method. The object contains a matrix of similarity indices between clusters, and the summaries of clusters extracted at first and second clustering.
`listclus`	`listclus` is a list of nested lists containing clustering solutions $\xi_1, \xi_2, ..., \xi_n$ at time points {`t1`, `t2`, $\cdots$, `tn`} respectively, and having the same length as

the number of time points. The `i^th` element of `listclus` is a nested list that contain set of clusters as matrices at corresponding time point `t_i` i.e. $\xi_i = X1, X2, , Xki$. For more details, *see* **Examples**.

`typeind` Type indicator. `typeind = 1` indicates that the raw data is provided in `listdata` argument, `typeind = 2` indicates that the `OverLap` objects are provided, whereas `typeind = 3` indicates that list of clusters are provided using `listclus` argument.

`Survival_thrHold`
A numeric value (0,1) indicating minimum threshold value for survival of clusters.

`Split_thrHold` A numeric value (0,1) indicating minimum threshold value for split of clusters.

`location_thrHold`
A numeric value (0,1) indicating minimum threshold value for shift in location of survived clusters.

`density_thrHold`
A numeric value (0,1) indicating minimum threshold value for changes in density of Survived clusters.

`k` Numeric Vector of length `vector("numeric", length = n-swSize+2)`. In the case of landmark window, its length is `n`, whereas in case of sliding window model its length is `n-swSize+2`, where `n` is the number of time points and `swSize` is the size of the sliding window. This argument should only be provided if `listdata` argument is chosen.

Details

The `Transition()` function apply 'MONIC' algorithm presented by Spiliopoulou et.al (2006) to trace changes in cluster solutions of dynamic data sets. The changes includes two types of transition i.e. External transition and Internal transition. External Transition consist of 'Survive', 'Split', 'Merge', 'Disappeared' and 'newly emerged' candidates, while Internal transition consist of changes in location and cohesion of the survived clusters. The `listdata` argument allow user to import dynamic datasets as a list of matrices or data frames, where each element of the list is a matrix containing data set at a single time point. Each dataset are clustered by 'kmeans' algorithm using default settings of `cclust()` function from `flexclust` package. The number of clusters at each time stamp can be import by `k` argument of the function, which is a vector of integers encompassing number of partitions in corresponding datasets of `listdata` argument. Once the datasets are clustered, the 'Overlap' matrices in clustering at consecutive time stamps are calculated. The Overlap matrix is calculated by using algorithm presented by Ntoutsi, I., et.al (2012). These 'Overlap' matrices are used to trace the transitions occurred in cluster solutions. Alternatively, the user can directly import list of 'Overlap' matrices between consecutive clustering. The Overlap matrix can be calculated using `Overlap(obj, e1, e2)` method of the package, where 'obj' is the object of class `OverLap` and e1, e2 are any clustering at time stamp i and j respectively. As a third option user can provide list of clusters at each data point utilizing `listclus` argument. Each element of the `listclus` is a nested list, which holds clusters at a single time stamp.

Value

Returns A list of class `Monic`.

Survive	Number of clusters survived.
Merged	Number of clusters merged.
Split	Number of clusters split.
Died	Number of clusters disappeared.
new.Emerged	Number of newly emerged clusters, which are not upshot of any external transition.
SurvivalCanx	A vector of integers indicating candidates from the first clustering survived to the latter time stamp
SurvivalCany	A vector of integers indicating candidates of second clustering, that clinch the survival candidates from first clustering.
SplitCanx	A vector of integers indicating candidate(s) that split into various daughter clusters from first clustering.
SplitCany	List of integer vector(s) designating candidates appeared, as a result of splits from first clustering.
MergeCanx	List of integer vector(s) designating Candidates that spliced together to form new clusters. Each element of the list gives candidates that merge together to form one.
MergeCany	Vector of integers designating candidates that emerged, as a result of merger of different candidates from first clustering.
EmergCan	Vector of integers contain Newly emerged candidates, which are not result of any external transition.
SurvivalRatio	The Ratio of survived clusters at second clustering to the total number of clusters at first clustering.
AbsorptionRatio	Ratio of number of merged clusters to total number of clusters at first clustering.
passforwardRatio	Sum of SurvivalRatio and AbsorptionRatio. This gives the ratio of clusters that is also present at second clustering either in the form of survival or absorption.
Overlap	A numeric matrix containing overlap of the two clustering. The rows of matrix indicate first clustering, while columns indicate second clustering.
Centersx	A matrix of cluster centers from first clustering.
Centersx	A matrix of cluster centers from second clustering.
rx	A numeric vector containing radius of each cluster from first clustering.
ry	A numeric vector containing radius of each cluster from second clustering.
avgDisx	A numeric vector containing average distance of points in a cluster from its center in first clustering.
avgDisy	A numeric vector containing average distance of points in a cluster from its center in second clustering.
ShiftLocCan	A vector of integers comprises of Survived candidates with shift in location.
NoShiftLocCan	A vector of integers comprises of Survived candidates with no shift in location.
MoreCompactCan	A Vector of integers comprises of Survived candidates, which becomes more compact.

`MoreDiffuseCan` A Vector of integers comprises of Survived candidates, which becomes more diffuse.

`NoChangeCompactCan`
A Vector of integers comprises of Survived candidates, with no changes in compactness.

`Location.diff` A numeric vector containing Distance between the centers of survived clusters.

`Compactness.diff`
A numeric vector containing Difference between compactness of survived clusters.

`Cluster_Tracex` A vector containing result of each cluster from first clustering.

`Cluster_Tracey` A Vector representing result of each cluster from second clustering.

`clusterMem` A vector of integers (from 1 to k) indicating the point to which cluster it is allocated from second clusterig.

References

Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y., Schult, R. MONIC: modeling and monitoring cluster transitions. In: Eliassi-Rad, T., Ungar, L. H., Craven, M., Gunopulos, D. (eds.) ACM SIGKDD 2006, pp. 706-711. ACM, Philadelphia (2006).

Examples

```
### Example 1: typeind = 1 (listdata Argument)

d1 <- Data2D[[1]][c("X1", "X2")]
d2 <- Data2D[[2]][c("X1", "X2")]
d3 <- Data2D[[3]][c("X1", "X2")]

listdata <- list(d1, d2, d3)

p <- Transition(listdata = listdata, swSize = 1, typeind = 1, Survival_thrHold = 0.8,
          Split_thrHold = 0.3, density_thrHold = 0.3, location_thrHold = 0.3, k = c(3,3,2))

### Example 2: typeind = 3 (listclus Argument)

D1 <- d1
D2 <- merge(d1, d2, all.x = TRUE, all.y = TRUE)
D3 <- merge(D2, d3, all.x = TRUE, all.y = TRUE)

set.seed(10)
f1 <- kmeans(D1, 3)
C1 <- list()
for(i in 1:3)C1[[i]] <- D1[f1$cluster == i, ]
f2 <- kmeans(D2, 3)
C2 <- list()
for(i in 1:3)C2[[i]] <- D2[f2$cluster == i, ]
f3 <- kmeans(D3, 2)
C3 <- list()
for(i in 1:2)C3[[i]] <- D3[f3$cluster == i, ]
```

```
listclus <- list(C1, C2, C3)

p <- Transition(listclus = listclus, typeind = 3, Survival_thrHold = 0.8,
                Split_thrHold = 0.3, density_thrHold = 0.3, location_thrHold = 0.3)

### Example 3: typeind = 3 (Overlap Argument)

obj <- new("OverLap")
Overlap1 <- Overlap(obj, e1 = C1, e2 = C2)
Overlap2 <- Overlap(obj, e1 = C2, e2 = C3)

Overlap <- list(Overlap1, Overlap2)
p <- Transition(Overlap = Overlap, typeind = 2, Survival_thrHold = 0.8,
                Split_thrHold = 0.3, density_thrHold = 0.3, location_thrHold = 0.3)
```

`TransitionCan-class` *External Transition Candidates*

Description

Class containing candidates that adopted external transition from first clustering ξ_1, and emerged as new clusters at second clustering ξ_2.

Slots

`SurvivalCanx` Vector of integers comprising Candidates that Survive from first clustering ξ_1.

`SurvivalCany` Vector of integers comprising Candidates that Survive to second clustering ξ_2.

`SplitCanx` Vector of integers comprising Candidates that Sliced into Various daughter Clusters from first clustering ξ_1.

`SplitCany` List of integer vectors comprising Candidates that emerged as daughter clusters in second clustering ξ_2 because of Split from first clustering ξ_1.

`MergeCanx` List of integer vectors comprising Candidates from first clustering ξ_1 that are merged. Each slot of list indicates the clusters that merge together from first clustering.

`MergeCany` Vector of integers comprising Candidates that emerged in second clustering ξ_2 because of merging various clusters from first clustering ξ_1.

`EmergCan` Newley emerged candidates which are not a result of any external transition from first clustering ξ_1.

`Cluster_Tracey` Vector of Cluster Trace from second clustering ξ_2.

`TransitionCount-class` *External Transition Count*

Description

Trace cluster solutions of dynamic datasets at consecutive time points and counts the clusters that experiences external transition. External transition includes Survive, Split, Merge, newly emerged, and Died candidates.

Slots

`Survive` Number of candidates survive from first clustering ξ_1.

`Split` Number of candidates from first clustering ξ_1 that split into several daughter clusters at second clustering ξ_2.

`Merge` Number of candidates from first clustering ξ_1 that merge toghter at second clustering ξ_2.

`Died` Number of candidates from first clusterin ξ_1 that disapeared at second clustering ξ_2.

`SurvivalRatio` Ratio of survive clusters to total number of clusters from first clusering ξ_1.

`AbsorptionRatio` Ratio of Merged clusters to total number of clusters from first clusering ξ_1.

`passforwardRatio` Sum of SurvivalRatio and AbsorptionRatio.

`Survival_thrHold` Threshold for survival of clusters.

`Split_thrHold` Threhold for split of clusters.

`Cluster_Tracex` Vector containing each cluster result from first clustering ξ_1.

Index