Package ‘longmixr’

October 13, 2022

Title Longitudinal Consensus Clustering with 'flexmix'

Version 1.0.0

Description An adaption of the consensus clustering approach from 'ConsensusClusterPlus' for longitudinal data. The longitudinal data is clustered with flexible mixture models from 'flexmix', while the consensus matrices are hierarchically clustered as in 'ConsensusClusterPlus'. By using the flexibility from 'flexmix' and 'FactoMineR', one can use mixed data types for the clustering.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

URL https://cellmapslab.github.io/longmixr/

BugReports https://github.com/cellmapslab/longmixr/issues

Depends R (>= 3.5.0)

biocViews

Imports checkmate, ConsensusClusterPlus, graphics, grDevices, flexmix, StatMatch, stats, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown, dplyr, tidyr, ggplot2, ggalluvial, FactoMineR, factoextra, lme4, purrr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Jonas Hagenberg [aut, cre] (<https://orcid.org/0000-0002-1849-1106>), Matt Wilkerson [aut, cph], Peter Waltman [aut, cph], Max Planck Institute of Psychiatry [cph]

Maintainer Jonas Hagenberg <jonas_hagenberg@psych.mpg.de>

Repository CRAN

Date/Publication 2022-01-13 20:32:42 UTC
R topics documented:

crosssectional_consensus_cluster .............................................. 2
fake_questionnaire_data .......................................................... 3
get_clusters ............................................................................ 4
longitudinal_consensus_cluster ..................................................... 5
plot.lcc ..................................................................................... 7
test_clustering_methods ............................................................... 8

Index 10

crosssectional_consensus_cluster

Cross-sectional clustering with categorical variables

Description

This function uses the ConsensusClusterPlus function from the package with the same name with defaults for clustering data with categorical variables. As the distance function, the Gower distance is used.

Usage

crosssectional_consensus_cluster(
  data,
  reps = 1000,
  finalLinkage = "ward.D2",
  innerLinkage = "ward.D2",
  ...
)

Arguments

data a matrix or data.frame containing variables that should be used for computing the distance. This argument is passed to StatMatch::gower.dist
reps number of repetitions, same as in ConsensusClusterPlus
finalLinkage linkage method for final clustering, same as in ConsensusClusterPlussame as in ConsensusClusterPlus
innerLinkage linkage method for clustering steps, same as in ConsensusClusterPlus
...
other arguments passed to ConsensusClusterPlus, attention: the d argument can not be set as it is directly computed by crosssectional_consensus_cluster

Details

data can take all input data types that gower.dist can handle, i.e. numeric, character/factor, ordered and logical.
Value

The output is produced by ConsensusClusterPlus

Examples

dc <- mtcars
# scale continuous variables
dc <- sapply(mtcars[, 1:7], scale)
# code factor variables
dc <- cbind(as.data.frame(dc),
    vs = as.factor(mtcars$vs),
    am = as.factor(mtcars$am),
    gear = as.factor(mtcars$gear),
    carb = as.factor(mtcars$carb))
cc <- crosssectional_consensus_cluster(
    data = dc,
    reps = 10,
    seed = 1
)

fake_questionnaire_data

Fake questionnaire data

Description

A simulated data set containing observations of 100 individuals at four time points. The data was simulated in two groups (50 individuals each) and contains two questionnaires with five items each, one questionnaire with five continuous variables and one additional cross-sectional continuous variable. In this data set the group variable from the simulation is included. You typically don’t have this group variable in your data.

Usage

fake_questionnaire_data

Format

A data frame with 400 rows and 20 variables:

ID patient ID
visit time point of the observation
group to which simulated group the observation belongs to
age_visit_1 age of the patient at time point 1
single_continuous_variable a cross-sectional continuous variable, i.e. there is only one unique value per individual
questionnaire_A_1 the first item of questionnaire A with categories 1 to 5
get_clusters

questionnaire_A_2 the second item of questionnaire A with categories 1 to 5
questionnaire_A_3 the third item of questionnaire A with categories 1 to 5
questionnaire_A_4 the fourth item of questionnaire A with categories 1 to 5
questionnaire_A_5 the fifth item of questionnaire A with categories 1 to 5
questionnaire_B_1 the first item of questionnaire B with categories 1 to 5
questionnaire_B_2 the second item of questionnaire B with categories 1 to 5
questionnaire_B_3 the third item of questionnaire B with categories 1 to 5
questionnaire_B_4 the fourth item of questionnaire B with categories 1 to 5
questionnaire_B_5 the fifth item of questionnaire B with categories 1 to 5
questionnaire_C_1 the first continuous variable of questionnaire C
questionnaire_C_2 the second continuous variable of questionnaire C
questionnaire_C_3 the third continuous variable of questionnaire C
questionnaire_C_4 the fourth continuous variable of questionnaire C
questionnaire_C_5 the fifth continuous variable of questionnaire C

Source
simulated data

---

get_clusters  
Extract the cluster assignments

Description
This function extracts the cluster assignments from an lcc object. One can specify which number of clusters the assignments should be returned.

Usage
get_clusters(cluster_solution, number_clusters = NULL)

Arguments
cluster_solution
    an lcc object
number_clusters
    default is NULL to return all assignments. Otherwise specify a numeric vector with the number of clusters for which the assignments should be returned, e.g. 2:4
Value

a data.frame with an ID column (the name of the ID column was specified by the user when calling the longitudinal_consensus_cluster function and one column with cluster assignments for every specified number of clusters. Only the assignments included in number_clusters are returned in the form of columns with the names assignment_num_clus_x.

Examples

# not run
set.seed(5)
test_data <- data.frame(patient_id = rep(1:10, each = 4),
visit = rep(1:4, 10),
var_1 = c(rnorm(20, -1), rnorm(20, 3)) +
rep(seq(from = 0, to = 1.5, length.out = 4), 10),
var_2 = c(rnorm(20, 0.5, 1.5), rnorm(20, -2, 0.3)) +
rep(seq(from = 1.5, to = 0, length.out = 4), 10))
model_list <- list(flexmix::FLXMRmgcv(as.formula("var_1 ~ .")),
flexmix::FLXMRmgcv(as.formula("var_2 ~ .")))
clustering <- longitudinal_consensus_cluster(
data = test_data,
id_column = "patient_id",
max_k = 2,
reps = 3,
model_list = model_list,
flexmix_formula = as.formula("~s(visit, k = 4) | patient_id"))
cluster_assignments <- get_clusters(clustering, number_clusters = 2)
# end not run
flexmix_formula = as.formula("~s(visit, k = 4) | patient_id"),
title = "untitled_consensus_cluster",
final_linkage = c("average", "ward.D", "ward.D2", "single", "complete", "mcquitty",
                 "median", "centroid"),
seed = 3794,
verbose = FALSE
)

Arguments

data a data.frame with one or several observations per subject. It needs to contain
one column that specifies to which subject the entry (row) belongs to. This ID
column is specified in id_column. Otherwise, there are no restrictions on the
column names, as the model is specified in flexmix_formula.

id_column name (character vector) of the ID column in data to identify all observations of
one subject

max_k maximum number of clusters, default is 3

reps number of repetitions, default is 10

p_item fraction of samples contained in subsampled sample, default is 0.8

model_list either one flexmix driver or a list of flexmix drivers of class FLXMR

flexmix_formula a formula object that describes the flexmix model relative to the formula in
the flexmix drivers (the dot in the flexmix drivers is replaced, see the example).
That means that you usually only specify the right-hand side of the formula here. However, this is not enforced or checked to give you more flexibility over the
flexmix interface

title name of the clustering; used if writeTable = TRUE

final_linkage linkage used for the last hierarchical clustering step on the consensus matrix; has
to be average, ward.D, ward.D2, single, complete, mcquitty, median or
centroid. The default is average

seed seed for reproducibility

verbose boolean if status messages should be displayed. Default is FALSE

Details

The data types longitudinal_consensus_cluster can handle depends on how the flexmix models are set up, in principle all data types are supported for which there is a flexmix driver with the
desired outcome variable.

If you follow the dimension reduction approach outlined in vignette("Example clustering analysis",
package = "longmixr"), the input data types depend on what FAMD from the FactoMineR package
can handle. FAMD accepts numeric variables and treats all other variables as factor variables which
it can handle as well.

Value

An object (list) of class lcc with length maxk. The first entry general_information contains the
entries:
consensus_matrices a list of all consensus matrices (for all specified clusters)

cluster_assignments a data.frame with an ID column named after id_column and a column for every specified number of clusters, e.g. assignment_num_clus_2

call the call/all arguments how longitudinal_consensus_cluster was called

The other entries correspond to the number of specified clusters (e.g. the second entry corresponds to 2 specified clusters) and each contains a list with the following entries:

consensus_matrix the consensus matrix

consensus_tree the result of the hierarchical clustering on the consensus matrix

consensus_class the resulting class for every observation

found_flexmix_clusters a vector of the actual found number of clusters by flexmix (which can deviate from the specified number)

Examples

```
set.seed(5)
test_data <- data.frame(patient_id = rep(1:10, each = 4),
                        visit = rep(1:4, 10),
                        var_1 = c(rnorm(20, -1), rnorm(20, 3)) +
                                rep(seq(from = 0, to = 1.5, length.out = 4), 10),
                        var_2 = c(rnorm(20, 0.5, 1.5), rnorm(20, -2, 0.3)) +
                                rep(seq(from = 1.5, to = 0, length.out = 4), 10))
model_list <- list(flexmix::FLXMRmgcv(as.formula("var_1 ~ .")),
                   flexmix::FLXMRmgcv(as.formula("var_2 ~ .")))
clustering <- longitudinal_consensus_cluster(
  data = test_data,
  id_column = "patient_id",
  max_k = 2,
  reps = 3,
  model_list = model_list,
  flexmix_formula = as.formula("~s(visit, k = 4) | patient_id"))
# not run
# plot(clustering)
# end not run
```

### plot.lcc

**Plot a longitudinal consensus clustering**

### Description

Plot a longitudinal consensus clustering

### Usage

```
# S3 method for class 'lcc'
plot(x, color_palette = NULL, ...)
```
**Arguments**

- `x`: lcc object (output from `longitudinal_consensus_cluster`)
- `color_palette`: optional character vector of colors for consensus matrix
- `...`: additional parameters for plotting; currently not used

**Value**

Plots the following plots:

- **consensus matrix legend**: the legend for the following consensus matrix plots
- **consensus matrix plot**: for every specified number of clusters, a heatmap of the consensus matrix and the result of the final clustering is shown
- **consensus CDF**: a line plot of the CDFs for all different specified numbers of clusters
- **Delta area**: elbow plot of the difference in the CDFs between the different numbers of clusters
- **tracking plot**: cluster assignment of the subjects throughout the different cluster solutions
- **item-consensus**: for every item (subject), calculate the average consensus value with all items that are assigned to one consensus cluster
- **cluster-consensus**: every bar represents the average pair-wise item-consensus within one consensus cluster

---

**test_clustering_methods**

*Try out different linkage methods*

**Description**

In the final step, the consensus clustering performs a hierarchical clustering step on the consensus cluster. This function tries out different linkage methods and returns the corresponding clusterings. The outputs can be plotted like the results from `longitudinal_consensus_cluster`.

**Usage**

```r
test_clustering_methods(
  results,
  use_methods = c("average", "ward.D", "ward.D2", "single", "complete", "mcquitty", "median", "centroid")
)
```

**Arguments**

- `results`: clustering result of class lcc
- `use_methods`: character vector of one or several items of average, ward.D, ward.D2, single, complete, mcquitty, median or centroid
Value

a list of elements, each element of class `lcc`. The entries are named after the used linkage method.

Examples

```r
set.seed(5)
test_data <- data.frame(patient_id = rep(1:10, each = 4),
visit = rep(1:4, 10),
var_1 = c(rnorm(20, -1), rnorm(20, 3)) +
rep(seq(from = 0, to = 1.5, length.out = 4), 10),
var_2 = c(rnorm(20, 0.5, 1.5), rnorm(20, -2, 0.3)) +
rep(seq(from = 1.5, to = 0, length.out = 4), 10))
model_list <- list(flexmix::FLXMRmgcv(as.formula("var_1 ~ .")),
flexmix::FLXMRmgcv(as.formula("var_2 ~ .")))
clustering <- longitudinal_consensus_cluster(
data = test_data,
id_column = "patient_id",
max_k = 2,
reps = 3,
model_list = model_list,
flexmix_formula = as.formula("~s(visit, k = 4) | patient_id"))
clustering_linkage <- test_clustering_methods(results = clustering,
use_methods = c("average", "single"))
# not run
# plot(clustering_linkage[["single"]]))
# end not run
```
Index

* datasets
  fake_questionnaire_data, 3
  crosssectional_consensus_cluster, 2
  fake_questionnaire_data, 3
  get_clusters, 4
  gower.dist, 2
  longitudinal_consensus_cluster, 5, 8
  plot.lcc, 7
  test_clustering_methods, 8