Package 'clustra'

January 16, 2022

Version 0.1.6

Date 2022-01-15

Title Clustering Longitudinal Trajectories

Depends R (>= 3.5.0)

Imports data.table, graphics, grDevices, methods, mgcv, MixSim, parallel, stats

Suggests ggplot2, knitr, rmarkdown

Description Clusters longitudinal trajectories over time (can be unequally spaced, unequal length time series and/or partially overlapping series) on a common time axis. Performs k-means clustering on a single continuous variable measured over time, where each mean is defined by a thin plate spline fit to all points in a cluster. Distance is MSE across trajectory points to cluster spline. Provides graphs of derived cluster splines, silhouette plots, and Adjusted Rand Index evaluations of the number of clusters. Scales well to large data with multicore parallelism available to speed computation.

LazyLoad yes

License BSD 2-clause License + file LICENSE

Encoding UTF-8

Maintainer George Ostrouchov <ostrouchovg@ornl.gov>

RoxygenNote 7.1.2

VignetteBuilder knitr

NeedsCompilation no

Author George Ostrouchov [aut, cre],
David Gagnon [aut],
Hanna Gerlovin [aut],
Chen Wei-Chen [ctb],
Schmidt Drew [ctb],
Oak Ridge National Laboratory [cph],
U.S. Department of Veteran's Affairs [fnd] (Project: Million Veteran Program Data Core)

Repository CRAN

Date/Publication 2022-01-16 06:42:41 UTC

R topics documented:

`clustra-package` *clustra-package*

Description

Clusters medical trajectories (unequally spaced and unequal lengths) aligned by an intervention time. Performs k-means clustering, where each mean is a thin plate spline fit to all points in a cluster. Distance is MSE across trajectory points to cluster spline. Provides silhouette plots and Adjusted Rand Index evaluations of the number of clusters. Scales well to large data with multicore parallelism available to speed computation.

Author(s)

George Ostrouchov, David Gagnon, Hanna Gerlovin

allpair_RandIndex *allpair_RandIndex: helper for replicated cluster comparison*

Description

Runs RandIndex for all pairs of cluster results in its list input and produces a matrix for use by rand_plot. Understands replicates within k values.

Usage

```
allpair_RandIndex(results)
```

Arguments

results	A list with each element packed internally by the clustra_rand function with elements: • k - number of clusters • rep - replicate number • deviance - final deviance • group - integer cluster assignments

Value

A data frame with RandIndex for all pairs from trajectories results. The data frame names and its format is intended to be the input for rand_plot. Note that all pairs is the lower triangle plus diagonal of an all-pairs symmetric matrix.

check_df *Checks if non-empty groups have enough data for spline fit degrees of freedom.*

Description

Checks if non-empty groups have enough data for spline fit degrees of freedom.

Usage

```
check_df(group, loss, data, maxdf)
```

Arguments

group	An integer vector of group membership for each id.
loss	A matrix with rows of computed loss values of each id across all models as columns.
data	A data.table with data. See trajectories.
maxdf	Fitting parameters. See trajectories.

Details

When a group has insufficient data for `maxdf`, its nearest model loss values are set to `Inf`, and new nearest model is assigned. The check repeats until all groups have sufficient data.

Value

Returns the vector of group membership of id's either unchanged or changed to have sufficient data in non-zero groups.

`clustra` *Cluster longitudinal trajectories over time*

Description

The usual top level function for clustering longitudinal trajectories. After initial setup, it calls `trajectories` to perform k-means clustering on continuous `response` measured over `time`, where each mean is defined by a thin plate spline fit to all points in a cluster. See `clustra_vignette.Rmd` for examples of use.

Usage

```
clustra(
  data,
  k,
  starts = c(1, 0),
  group = NULL,
  maxdf = 30,
  conv = c(10, 0),
  mccores = 1,
  verbose = FALSE
)
```

Arguments

`data`	Data frame or, preferably, also a data.table with response measurements, one response per observation. Required variables are (id, time, response). Other variables are ignored.
`k`	Number of clusters
`starts`	A vector of length two. See `start_groups`.
`group`	A vector of initial cluster assignments for unique id's in data. Normally, this is NULL and good starts are provided by `start_groups`.
`maxdf`	Fitting parameters. See `trajectories`.
`conv`	Fitting parameters. See `trajectories`.
`mccores`	See `trajectories`.
`verbose`	Logical to turn on more output during fit iterations.

Value

A list returned by `trajectories` plus one more element `ido`, giving the original id numbers.

Examples

```
set.seed(13)
data = gen_traj_data(n_id = c(50, 100), m_obs = 20, s_range = c(-365, -14),
              e_range = c(0.5*365, 2*365))
cl = clustra(data, k = 2, maxdf = 20, conv = c(5, 0), verbose = TRUE)
tabulate(data$group)
tabulate(data$true_group)
```

`clustra_rand` *clustra_rand: Rand Index cluster evaluation*

Description

Performs `trajectories` runs for several k and several random start replicates within k. Returns a data frame with a Rand Index comparison between all pairs of clusterings. This data frame is typically input to `rand_plot` to produce a heat map with the Adjusted Rand Index results.

Usage

```
clustra_rand(
  data,
  k,
  mccores,
  replicates = 10,
  maxdf = 30,
  conv = c(10, 0),
  save = FALSE,
  verbose = FALSE
)
```

Arguments

`data`	The data (see `clustra` description).
`k`	Vector of k values to try.
`mccores`	Number of cores for replicate parallelism via mclapply.
`replicates`	Number of replicates for each k.
`maxdf`	Fitting parameters. See `link{trajectories}`.
`conv`	Fitting parameters. See `link{trajectories}`.
`save`	Logical. When TRUE, save all results as file `results.Rdata`.
`verbose`	Logical. When TRUE, information about each run of clustra is printed.

Value

See allpair_RandIndex.

---

clustra_sil *clustra_sil: Prepare silhouette plot data for several k or for a previous clustra run*

Description

Performs clustra runs for several k and prepares silhouette plot data. Computes a proxy silhouette index based on distances to cluster centers rather than trajectory pairs. The cost is essentially that of running clustra for several k as this information is available directly from clustra. Can also reuse a previous clustra run and produce data for a single silhouette plot.

Usage

```
clustra_sil(
  data,
  k = NULL,
  mccores = 1,
  maxdf = 30,
  conv = c(10, 0),
  save = FALSE,
  verbose = FALSE
)
```

Arguments

data	Either a data.frame (data parameter of trajectories) or the output from a clustra run. See Details.
k	Vector of k values to try. If output from clustra is the data parameter, k can be left NULL or set to the number of clusters used.
mccores	See trajectories.
maxdf	Fitting parameters. See trajectories.
conv	Fitting parameters. See trajectories.
save	Logical. When TRUE, save all results as file clustra_sil.Rdata.
verbose	Logical. When TRUE, information about each run of clustra is printed.

Details

When given the raw data as the first parameter (input data parameter of trajectories), k can also specify a vector of cluster numbers to run clustra and then produce silhouette plots for each of them. Alternatively, the input can be the output from a clustra run, in which case data for a single silhouette plot will be made without rerunning clustra.

Value

Invisibly returns a list of length `length(k)`, where each element is a matrix with `nrow(data)` rows and three columns `cluster`, `neighbor`, `silhouette`. This list of matrices can be used to draw a silhouette plot.

---

deltime *Timing function*

Description

Timing function

Usage

```
deltime(ltime = proc.time()["elapsed"], text = NULL)
```

Arguments

`ltime`	Result of last call to deltime.
`text`	Text to display along with elapsed time.

Value

"elapsed" component of current `proc.time`.

---

gendata *gendata*

Description

Generates data for up to three trajectory clusters

Usage

```
gendata(n_id, m_obs, s_range, e_range, min_obs, reference, noise)
```

Arguments

`n_id`	See parameters of `gen_traj_data`.
`m_obs`	See parameters of `gen_traj_data`.
`s_range`	See parameters of `gen_traj_data`.
`e_range`	See parameters of `gen_traj_data`.
`min_obs`	See parameters of `gen_traj_data`.
`reference`	See parameters of `gen_traj_data`.
`noise`	See parameters of `gen_traj_data`.

Details

Time support of each `id` is at least s . . . 0 . . . . e, where `s` is in `s_range` and `e` is in `e_range`.

Value

A list of length `sum(n_id)`, where each element is a matrix output by `oneid`.

`gen_traj_data` *Data Generators*

Description

Generates a collection of longitudinal responses with possibly varying lengths and varying numbers of observations. Support is `start . . . 0 . . . end`, where `start`~uniform(s_range) and `end`~uniform(e_range), so that all trajectories are aligned at 0 but can start and end at different times. Zero is the intervention time.

Usage

```
gen_traj_data(
  n_id,
  m_obs,
  s_range,
  e_range,
  reference = 100,
  noise = c(0, abs(reference/20)),
  min_obs = 3
)
```

Arguments

`n_id`	Vector whose length is the number of clusters, giving the number of id's to generate in each cluster.
`m_obs`	Mean number of observation per id. Provides `lambda` parameter in `rpois`.
`s_range`	A vector of length 2, giving the min and max limits of uniformly generated start observation time.
`e_range`	A vector of length 2, giving the min and max limits of uniformly generated end observation time.
`reference`	A nominal response value (for example, blood pressure is near 100, which is the default)
`noise`	Vector of length 2 giving the *mean* and *sd* of added N(mean, sd) noise.
`min_obs`	Minimum number of observations in addition to zero time observation.

Value

A data table with one response per row and four columns: `id`, `time`, `response`, and `true_group`.

Details

Generate longitudinal data for a response variable. Trajectories start at time uniformly distributed in s_range and end at time uniformly distributed in e_range. Number of observations in a trajectory is Poisson(m_obs). The result is a number of trajectories, all starting at time 0, with different time spans, and with independently different numbers of observations within the time spans. Each trajectory follows a randomly selected response function with added N(mean, sd) error.

Examples

```
data = gen_traj_data(n_id = c(50, 100), m_obs = 20, s_range = c(-365, -14),
              e_range = c(0.5*365, 2*365))
head(data)
tail(data)
```

---

mse_g *Loss functions*

Description

`mse_g()` Computes mean-squared error. `mxe_g()` Computes maximum absolute error.

Usage

```
mse_g(pred, id, response)

mxe_g(pred, id, response)
```

Arguments

`pred`	Vector of predicted values.
`id`	Integer vector of group assignments.
`response`	Vector of response values.

Value

A numeric value. For mse_g(), returns the mean-squared error. For mxe_g(), returns the maximum absolute error.

oneid *Generates data for one* id

Description

Generates data for one id

Usage

```
oneid(id, n_obs, type, start, end, smin, emax, reference, noise)
```

Arguments

id	A unique integer.
n_obs	An integer number of observations to produce.
type	Response type, 1 is constant, 2 is a sin curve portion, and 3 is a sigmoid portion.
start	Negative integer giving time of first observation.
end	Positive integer giving time of last observation.
smin	The smallest possible start value among all ids. Currently not used.
emax	The largest possible end value among all ids. Used to scale sin and sigmoid support.
reference	A response value for constant response. Also used to scale sin and sigmoid responses.
noise	Standard deviation of zero mean Gaussian noise added to response functions.

Value

An n_obs by 4 matrix with columns id, time, response, true_group.

pred_g *Function to predict for new data based on fitted gam object.*

Description

Function to predict for new data based on fitted gam object.

Usage

```
pred_g(tps, newdata)
```

Arguments

tps	Output structure of bam.
newdata	See clustra description of data parameter.

Value

A numeric vector of predicted values corresponding to rows of newdata. If gam object is NULL, NULL is returned instead.

`rand_plot` *Matrix plot of Rand Index comparison of replicated clusters*

Description

Matrix plot of Rand Index comparison of replicated clusters

Usage

```
rand_plot(rand_pairs, name = NULL)
```

Arguments

`rand_pairs`	A data frame result of `allpair_RandIndex`
`name`	Character string file name for pdf plot. If omitted or NULL, plot will render to current graphics device.

Value

Invisible. Full path name of file with plot.

Author(s)

Wei-Chen Chen and George Ostrouchov

References

Wei-chen Chen, George Ostrouchov, David Pugmire, Prabhat, and Michael Wehner. 2013. A Parallel EM Algorithm for Model-Based Clustering Applied to the Exploration of Large Spatio-Temporal Data. Technometrics, 55:4, 513-523.

Sorts replicates within cluster K Assumes K starts from 2

start_groups *Function to assign starting groups.*

Description

If only one start, a random assignment is done. If more than one start, picks tps fit with smallest deviance after one iteration among random starts. Choosing from samples increases diversity of fits (sum of distances between group fits). Then classifies all ids based on fit from best sample.

Usage

```
start_groups(data, k, starts, maxdf, conv, mccores = 1, verbose = FALSE)
```

Arguments

data	Data.table with response measurements, one per observation. Column names are id, time, response, group. Note that ids are assumed sequential starting from 1. This affects expanding group numbers to ids.
k	Number of clusters (groups).
starts	A vector of length two, giving the number of start values to try and the number of ids per cluster to evaluate the starts (If the number of ids is less than 1, use all data and do not subset data for starts.). The default is c(1,0), meaning that one random start is used with all the data. The following are experimental at this time: If more than one start is requested, the most diverse after one trajectories iteration on a sample of data is used. Diversity is measured as sum of pairwise distances between models on a time grid of 2*maxdf points.
maxdf	Fitting parameters. See trajectories.
conv	Fitting parameters. See trajectories.
mccores	See trajectories.
verbose	Turn on more output for debugging.

Value

An integer vector corresponding to unique ids, giving group number assignments.

tps_g *Fits a thin plate spline to a single group with bam.*

Description

Fits a thin plate spline to a single group (one list element) in data with bam. Uses data from only on group rather than a zero weights approach. Zero weights would result in incorrect crossvalidation sampling.

Usage

```
tps_g(g, data, maxdf, nthreads)
```

Arguments

g	Integer group number.
data	A list of group-separated data using lapply with data.table::copy(data[group == g]) from original data in clustra description.
maxdf	See trajectories description.
nthreads	Controls bam threads.

Value

Returns an object of class "gam". See bam value. If group data has zero rows, NULL is returned instead.

---

trajectories *Cluster longitudinal trajectories over time.*

Description

Performs k-means clustering on continuous response measured over time, where each mean is defined by a thin plate spline fit to all points in a cluster. Typically, this function is called by clustra.

Usage

```
trajectories(
  data,
  k,
  group,
  maxdf,
  conv = c(10, 0),
  mccores = 1,
  verbose = FALSE
)
```

Arguments

data	Data table or data frame with response measurements, one per observation. Column names are id, time, response, group. Note that ids must be sequential starting from 1. This affects expanding group numbers to ids.
k	Number of clusters (groups)
group	Vector of initial group numbers corresponding to ids.

maxdf	Integer. Basis dimension of smooth term. See s function parameter k, in package mgcv.
conv	A vector of length two, c(iter,minchange), where iter is the maximum number of EM iterations and minchange is the minimum percentage of subjects changing group to continue iterations. Setting minchange to zero continues iterations until no more changes occur or maxiter is reached.
mccores	Integer number of cores to use by mclapply sections. Parallelization is over k, the number of clusters.
verbose	Logical, whether to produce debug output.

Value

A list with components

- deviance - The final deviance in each cluster added across clusters.
- group - Integer vector of group assignments corresponding to unique ids.
- loss - Numeric matrix with rows corresponding to unique ids and one column for each cluster. Each entry is the mean squared loss for the data in the id relative to the cluster model.
- k - An integer giving the requested number of clusters.
- k_cl - An integer giving the converged number of clusters. Can be smaller than k when some clusters become too small for degrees of freedom during convergence.
- data_group - An integer vector, giving group assignment as expanded into all id time points.
- tps - A list with k_cl elements, each an object returned by the mgcv::bam fit of a cluster thin plate spline model.
- iterations - An integer giving the number of iterations taken.
- counts - An integer vector giving the number of ids in each cluster.
- counts_df - An integer vector giving the total number of observations in each cluster (sum of the number of observations for ids belonging to the cluster).
- changes - An integer, giving the number of ids that changed clusters in the last iteration. This is zero if converged.

Author(s)

George Ostrouchov and David Gagnon

---

traj_rep *Function to run trajectories inside mclapply with one core.*

Description

Function to run trajectories inside mclapply with one core.

Usage

```
traj_rep(group, data, k, maxdf, conv)
```

Arguments

group	Vector of starting group values for unique id's.
data	The data (see clustra description).
k	Integer number of clusters.
maxdf	Fitting parameters. See trajectories.
conv	Fitting parameters. See trajectories.

Value

See return of trajectories.

---

xit_report *xit_report*

Description

Examines trajectories output to name what was concluded, such as convergence, maximum iterations reached, a zero cluster, etc. Multiple conclusions are possible as not all are mutually exclusive.

Usage

```
xit_report(cl, maxdf, conv)
```

Arguments

cl	Output structure from trajectories function
maxdf	Fitting parameters. See trajectories.
conv	Fitting parameters. See trajectories.

Value

NULL or a character vector of exit criteria satisfied.

Index