Package ‘ssMRCD’

May 15, 2023

Type Package
Title Spatially Smoothed MRCD Estimator
Version 0.1.0
Maintainer Patricia Puchhammer <patricia.puchhammer@tuwien.ac.at>
Description Estimation of the Spatially Smoothed Minimum Regularized Determinant (ssMRCD) estimator and its usage in an ssMRCD-based outlier detection method as described in Puchhammer and Filzmoser (2023) <doi:10.48550/arXiv.2305.05371>. Included are also complementary visualization and parameter tuning tools.
License GPL-3
Encoding UTF-8
LazyData true
Imports stats, grDevices, graphics, robustbase, scales, car, dbscan, plot3D, dplyr, ggplot2
RoxygenNote 7.2.3
Suggests knitr, rmarkdown, testthat (>= 3.0.0)
Config/testthat/edition 3
Depends R (>= 4.0.0)
VignetteBuilder knitr
NeedsCompilation no
Author Patricia Puchhammer [aut, cre, cph], Peter Filzmoser [aut]
Repository CRAN
Date/Publication 2023-05-15 19:20:02 UTC

R topics documented:

- contamination_random .............................................. 2
- geo_weights .......................................................... 3
- local_outliers_ssMRCD ............................................. 4
- N_structure_gridbased ............................................. 5
**contamination_random**  

**Description**  
This function swaps observations completely random in order to introduce contamination in the data. Used in parameter_tuning.

**Usage**  

```r  
contamination_random(cont, data)  
```

**Arguments**  
- `cont` numeric, amount of contamination in data.  
- `data` data whose observations should be switched.

**Value**  
A matrix with switched observations.

**Examples**  
# set seed  
set.seed(1)  

# get data  
data(weatherAUT2021)  

# switch 5% of observations  
contamination_random(cont = 0.05, data = weatherAUT2021[,1:6])
geo_weights

Inverse Geographic Weight Matrix

Description

Calculates a inverse-distance based weight matrix for the function ssMRC (see details).

Usage

geo_weights(coordinates, N_assignments)

Arguments

coordinates matrix of coordinates of observations.
N_assignments vector of neighborhood assignments.

Details

First, the centers (means of the coordinates given) $c_i$ of each neighborhood is calculated. Then, the Euclidean distance between the centers is calculated and the weight is based on the inverse distance between two neighborhoods,

$$w_{ij} = \frac{1}{\text{dist}(c_i, c_j)}.$$  

It is scaled according to a weight matrix.

Value

Returns a weighting matrix $W$ and the coordinates of the centers per neighborhood $cN$.

See Also

rescale_weights

Examples

coordinates = matrix(rnorm(1000), ncol = 2, nrow = 500)
N_ass = sample(1:5, 500, replace = TRUE)

geo_weights(coordinates, N_ass)
Local Outlier Detection Technique based on ssMRCD

Description

This function applies the local outlier detection method based on the spatially smoothed MRCD estimator developed in Puchhammer and Filzmoser (2023).

Usage

```r
local_outliers_ssMRCD(
  data,
  coords,
  N_assignments,
  lambda,
  weights = NULL,
  k = NULL,
  dist = NULL
)
```

Arguments

data data matrix with measured values.
coords matrix of coordinates of observations.
N_assignments vector of neighborhood assignments.
lambda scalar used for spatial smoothing (see also `ssMRCD`).
weights weight matrix used in `ssMRCD`.
k integer, if given the k nearest neighbors per observations are used to calculate next distances. Default value is k = NULL.
dist scalar, if given the neighbors closer than given distance are used for next distances. If dist is given, dist is used, otherwise k is used.

Value

Returns an object of class "locOuts" with following components:

outliers indices of found outliers.
next_distance vector of next distances for all observations.
cutoff upper fence of adjusted boxplot (see `adjbox`) used as cutoff value for next distances.
coords matrix of observation coordinates.
data matrix of observation values.
N_structure_gridbased

N_assignments  vector of neighborhood assignments.
k, dist    specifications regarding neighbor comparisons.
centersN    coordinates of centers of neighborhoods.
matneighbor    matrix storing information which observations where used to calculate next distance for each observation (per row). 1 indicates it is used.
ssMRCD    object of class "ssMRCD" and output of ssMRCD covariance estimation.

References


See Also

See also functions ssMRCD, plot.locOuts, summary.locOuts.

Examples

# data construction
data = matrix(rnorm(2000), ncol = 4)
coords = matrix(rnorm(1000), ncol = 2)
N_assignments = sample(1:10, 500, replace = TRUE)
lambda = 0.3

# apply function
outs = local_outliers_ssMRCD(data = data,
                          coords = coords,
                          N_assignments = N_assignments,
                          lambda = lambda,
                          k = 10)
outs

N_structure_gridbased  Creates Grid-Based Neighborhood Structure

Description

This function creates a grid-based neighborhood structure for the ssMRCD function using cut-off values for two coordinate axis.

Usage

N_structure_gridbased(x, y, cutx, cuty)
Arguments

x  vector of first coordinate of data set.
y  vector of second coordinate of data set.
cutx  cut-offs for first coordinate.
cuty  cut-offs for second coordinate.

Value

Returns a neighborhood assignment vector for the coordinates x and y.

Examples

# get data
data(weatherAUT2021)

# set cut-off values
cut_lon = c(9:16, 18)
cut_lat = c(46, 47, 47.5, 48, 49)

# create neighborhood assignments
N_structure_gridbased(weatherAUT2021$lon,
                       weatherAUT2021$lat,
                       cut_lon,
                       cut_lat)

objective_matrix  Calculation of Objective Function

Description

Calculation of the value of the objective function for the ssMRCD for a given list of matrices, lambda and a weighting matrix according to formula (3) in Puchhammer and Filzmoser (2023).

Usage

objective_matrix(matrix_list, lambda, weights)

Arguments

matrix_list  a list of matrices $K_i$
lambda  scalar smoothing parameter
weights  matrix of weights

Value

Returns the value of the objective function using matrices $K_i$. 
References


Examples

```r
# construct matrices
k1 = matrix(c(1,2,3,4), nrow = 2)
k2 = matrix(c(1,3,5,7), nrow = 2)

# construct weighting matrix
W = matrix(c(0, 1, 1, 0), nrow = 2)

objective_matrix(list(k1, k2), 0.5, W)
```

parameter_tuning | Parameter Tuning

Description

This function provides insight into the effects of different parameter settings.

Usage

```r
parameter_tuning(
  data, 
  coords, 
  N_assignments, 
  lambda = c(0, 0.25, 0.5, 0.75, 0.9), 
  weights = NULL, 
  k = NULL, 
  dist = NULL, 
  cont = 0.05, 
  repetitions = 5
)
```

Arguments

data | matrix with observations.
coords | matrix of coordinates of these observations.
N_assignments | numeric vector, the neighborhood structure that should be used for ssMRCD.
lambda | scalar, the smoothing parameter.
weights | weighting matrix used in ssMRCD.
k | vector of possible k-values to evaluate.
dist | vector of possible dist-values to evaluate.
cont | level of contamination, between 0 and 1.
repetitions | number of repetitions wanted to have a good picture of the best parameter combination.
Value

Returns a matrix of average false-negative rate (FNR) values and the total number of outliers found by the method as a proxy for the false-positive rate. Be aware that the FNR does not take into account that there are also natural outliers included in the data set that might or might not be found. Also a plot is returned representing these average. The best parameter selection depends on the goal of the analysis.

Examples

```r
# get data set
data("weatherAUT2021")

# make neighborhood assignments
cut_lon = c(9:16, 18)
cut_lat = c(46, 47, 47.5, 48, 49)
N = ssMRCD::N_structure_gridbased(weatherAUT2021$lon, weatherAUT2021$lat, cut_lon, cut_lat)
table(N)
N[N == 2] = 1
N[N == 3] = 4
N[N == 5] = 4
N[N == 6] = 7
N[N == 11] = 15
N = as.numeric(as.factor(N))

# tune parameters
set.seed(123)
parameter_tuning(data = weatherAUT2021[, 1:6],
                 coords = weatherAUT2021[, c("lon", "lat")],
                 N_assignments = N,
                 lambda = c(0.5, 0.75),
                 k = c(10),
                 repetitions = 1)
```

---

**plot.locOuts**

*Diagnostic Plots for Local Outlier Detection*

**Description**

This function plots different diagnostic plots for local outlier detection. It can be applied to an object of class "locOuts" which is the output of the function `local_outliers_ssMRCD`.

**Usage**

```r
## S3 method for class 'locOuts'
plot(
    x,
    type = c("hist", "spatial", "lines", "3D"),
```
plot.locOuts

```r
plot.locOuts = function(x, type, colour = "all", focus = NULL, pos = NULL, alpha = 0.3, data = NULL, add_map = TRUE, ...)
```

**Arguments**

- `x` a locOuts object obtained by the function `local_outliers_ssMRCD`.
- `type` vector containing the types of plots that should be plotted, possible values c("hist", "spatial", "lines", "3D").
- `colour` character specifying the color scheme (see details). Possible values "all", "onlyOuts", "outScore".
- `focus` an integer being the index of the observation whose neighborhood should be analysed more closely.
- `pos` integer specifying the position of the text "cut-off" in the histogram (see `par`).
- `alpha` scalar specifying the transparency level of the points plotted for plot type "spatial", "3D" and "lines".
- `data` optional data frame or matrix used for plot of type "line". Will be used to plot lines based scaled data instead of the data used for local outlier detection.
- `add_map` TRUE if a map should be plotted along the line plot (type = "lines").
- `...` further parameters passed on to base-R plotting functions.

**Details**

Regarding the parameter `type` the value "hist" corresponds to a plot of the histogram of the next distances together with the used cutoff-value. When using "spatial" the coordinates of each observation are plotted and colorized according to the color setting. The "lines" plot is used with the index `focus` of one observation whose out/inlyingness to its neighborhood should by plotted. The whole data set is scaled to the range [0,1] and the scaled value of the selected observation and its neighbors are plotted. Outliers are plotted in orange. The "3D" setting leads to a 3D-plot using the colour setting as height. The view can be adapted using the parameters theta and phi.

For the `colour` setting possible values are "all" (all next distances are used and colored in an orange palette), "onlyOuts" (only outliers are plotted in orange, inliers are plotted in grey) and "outScore" (the next distance divided by the cutoff value is used to colourize the points; inliers are colorized in blue, outliers in orange).

**Value**

Returns plots regarding next distances and spatial context.

**See Also**

`local_outliers_ssMRCD`
Examples

# set seed
set.seed(1)

# make locOuts object
data = matrix(rnorm(2000), ncol = 4)
coords = matrix(rnorm(1000), ncol = 2)
N_assignments = sample(1:10, 500, replace = TRUE)
lambda = 0.3

# local outlier detection
outs = local_outliers_ssMRCD(data = data,
    coords = coords,
    N_assignments = N_assignments,
    lambda = lambda,
    k = 10)

# plot results
plot(outs, type = "hist")
plot(outs, type = "spatial", colour = "outScore")
plot(outs, type = "3D", colour = "outScore", theta = 0)
plot(outs, type = "lines", focus = outs$outliers[1])

---

plot.ssMRCD

Plot Method for ssMRCD Object

Description

Plots diagnostics for function output of ssMRCD regarding convergence behavior and the resulting covariances matrices.

Usage

```r
## S3 method for class 'ssMRCD'
plot(
x,
    type = c("convergence", "ellipses"),
    centersN = NULL,
    colour_scheme = "none",
    xlim_upper = 9,
    manual_rescale = 1,
    legend = TRUE,
    xlim = NULL,
    ylim = NULL,
    ...
)
```
Arguments

- **x**: object of class "ssMRCD".
- **type**: type of plot, possible values are "convergence" and "ellipses". See details.
- **centersN**: for plot type "ellipses" a matrix specifying the positions of the centers of the covariance estimation centers, see also `geo_weights`.
- **colour_scheme**: coloring scheme used for plot type "ellipses", either "trace" or "regularity" or "none".
- **xlim_upper**: numeric giving the upper x limit for plot type "convergence".
- **manual_rescale**: numeric used to re-scale ellipse sizes.
- **legend**: logical, if color legend should be included.
- **xlim**: vector of xlim (see `par`).
- **ylim**: vector of ylim (see `par`).
- **...**: further plotting parameters.

Details

For `type = "convergence"` a plot is produced displaying the convergence behaviour. Each line represents a different initial value used for the c-step iteration. On the x-axis the iteration step is plotted with the corresponding value of the objective function. Not monotonically lines are plotted in red.

For `type = "ellipses"` and more than a 2-dimensional data setting plotting the exact tolerance ellipse is not possible anymore. Instead the two eigenvectors with highest eigenvalue from the MCD used on the full data set without neighborhood assignments are taken and used as axis for the tolerance ellipses of the ssMRCD covariance estimators. The tolerance ellipse for the global MCD covariance is plotted in grey in the upper left corner. It is possible to set the colour scheme to "trace" to see the overall amount of variability and compare the plotted covariance and the real trace to see how much variance is not plotted. For "regularity" the regularization of each covariance is shown.

Value

Returns plots of the ssMRCD methodology and results.

See Also

- `ssMRCD`, `summary.ssMRCD`, `local_outliers_ssMRCD`, `plot.locOuts`

Examples

```r
# set seed
set.seed(1)

# create data set
data = matrix(rnorm(2000), ncol = 4)
coords = matrix(rnorm(1000), ncol = 2)
```
N_assignments = sample(1:10, 500, replace = TRUE)
lambda = 0.3

# calculate ssMRCD by using the local outlier detection method
outs = local_outliers_ssMRCD(data = data,
    coords = coords,
    N_assignments = N_assignments,
    lambda = lambda,
    k = 10)

# plot ssMRCD object included in outs
plot(x = outs$ssMRCD,
    centersN = outs$centersN,
    colour_scheme = "trace",
    legend = FALSE)

---

### rescale_weights

**Rescale Weight Matrix**

**Description**

Given a matrix with values for neighborhood influences the function rescales the matrix in order to get an appropriate weight matrix used for the function ssMRCD.

**Usage**

rescale_weights(W)

**Arguments**

- **W**
  - weight matrix with diagonals equal to zero and at least one positive entry per row.

**Value**

An appropriately scaled weight matrix.

**See Also**

ssMRCD, local_outliers_ssMRCD, geo_weights

**Examples**

```r
W = matrix(c(0, 1, 2,
             1, 0, 1,
             2, 1, 0), nrow = 3)
rescale_weights(W)
```
restructure_as_list

Restructure Data Matrix as List

Description

This function restructures neighborhood information given by a data matrix containing all information and one neighborhood assignment vector. It returns a list of data matrices used in ssMRCD.

Usage

restructure_as_list(data, neighborhood_vec)

Arguments

data data matrix with all observations.
neighborhood_vec numeric neighborhood assignment vector. Should contain numbers from 1 to N and not leave integers out.

Value

Returns a list containing the observations per neighborhood assignment.

Examples

# data matrix
data = matrix(rnorm(n = 3000), ncol = 3)
N_assign = sample(x = 1:10, size = 1000, replace = TRUE)
restructure_as_list(data, N_assign)

ssMRCD

Spatially Smoothed MRCD Estimator

Description

The ssMRCD function calculates the spatially smoothed MRCD estimator from Puchhammer and Filzmoser (2023).
Usage

ssMRCD(
  x,
  weights,
  lambda,
  TM = NULL,
  alpha = 0.75,
  maxcond = 50,
  maxcsteps = 200,
  n_initialhsets = NULL
)

Arguments

x    a list of matrices containing the observations per neighborhood sorted which can
     be obtained by the function restructure_as_list.
weights weighting matrix, symmetrical, rows sum up to one and diagonals need to be
     zero (see also geo_weights or rescale_weights.
lambda numeric between 0 and 1.
TM    target matrix (optional), default value is the covMcd from robustbase.
alpha numeric, proportion of values included, between 0.5 and 1.
maxcond optional, maximal condition number used for rho-estimation.
maxcsteps maximal number of c-steps before algorithm stops.
n_initialhsets number of initial h-sets, default is 6 times number of neighborhoods.

Value

An object of class "ssMRCD" containing the following elements:

MRCDcov List of ssMRCD-covariance matrices sorted by neighborhood.
MRCDicov List of inverse ssMRCD-covariance matrices sorted by neighborhood.
MRCDmu List of ssMRCD-mean vectors sorted by neighborhood.
mX List of data matrices sorted by neighborhood.
N Number of neighborhoods.
mT Target matrix.
rho Vector of regularization values sorted by neighborhood.
alpha Scalar what percentage of observations should be used.
h Vector of how many observations are used per neighborhood, sorted.
numiter  The number of iterations for the best initial h-set combination.
c_alpha  Consistency factor for normality.
weights  The weighting matrix.
lambda  Smoothing factor.
obj_fun_values  A matrix with objective function values for all initial h-set combinations (rows) and iterations (columns).
best6pack  initial h-set combinations with best objective function value after c-step iterations.
Kcov  returns MRCD-estimates without smoothing.

References


See Also

plot.ssMRCD, summary.ssMRCD, restructure_as_list

Examples

# create data set
x1 = matrix(runif(200), ncol = 2)
x2 = matrix(rnorm(200), ncol = 2)
x = list(x1, x2)

# create weighting matrix
W = matrix(c(0, 1, 1, 0), ncol = 2)

# calculate ssMRCD
ssMRCD(x, weights = W, lambda = 0.5)

summary.locOuts  Summary of Local Outlier Detection

Description

Prints a summary of the locOuts object obtained by the function local_outliers_ssMRCD.

Usage

## S3 method for class 'locOuts'
summary(object, ...)


Arguments

object

... further parameters passed on.

Value

Prints a summary of the locOuts object.

See Also

plot.locOuts

Examples

# set seed
set.seed(1)

# make locOuts object
data = matrix(rnorm(2000), ncol = 4)
coords = matrix(rnorm(1000), ncol = 2)
N_assignments = sample(1:10, 500, replace = TRUE)
lambda = 0.3

# local outlier detection
outs = local_outliers_ssMRCD(data = data,
   coords = coords,
   N_assignments = N_assignments,
   lambda = lambda,
   k = 10)

# summary method
summary(outs)
Value

Prints a summary of the ssMRCD object.

See Also

See also ssMRCD, plot.ssMRCD.

---

weatherAUT2021  Austrian Weather Data 2021

Description

This data is a subset of the GeoSphere Austria monthly weather data of 2021 averaged using the median. Stations with missing values are removed.

Usage

weatherAUT2021

Format

A data frame with 183 rows and 10 columns:

- **name**: Unique name of the weather station in German.
- **lon, lat**: Longitude and latitude of the weather station.
- **alt**: Altitude of the weather station (meter).
- **p**: Average air pressure (hPa).
- **s**: Monthly sum of sunshine duration (hours).
- **vv**: Wind velocity (meter/second).
- **t**: Air temperature in 2 meters above the ground in (°C).
- **rsum**: Average daily sum of precipitation (mm).
- **rel**: Relative air humidity (percent).

Source

The original data was downloaded here (December 2022): https://data.hub.zamg.ac.at/dataset/klima-v1-1m.

References

Data Source: GeoSphere Austria - https://data.hub.zamg.ac.at.

Examples

data(weatherAUT2021)
summary(weatherAUT2021)
Index

* datasets
  weatherAUT2021, 17

adjbox, 4
contamination_random, 2
geo_weights, 3, 11, 12, 14
local_outliers_ssMRCD, 4, 8, 9, 11, 12, 15
N_structure_gridbased, 5
objective_matrix, 6
par, 9, 11
parameter_tuning, 2, 7
plot.locOuts, 5, 8, 11, 16
plot.ssMRCD, 10, 15, 17
rescale_weights, 3, 12, 14
restructure_as_list, 13, 14, 15
ssMRCD, 3–7, 10–13, 13, 16, 17
summary.locOuts, 5, 15
summary.ssMRCD, 11, 15, 16
weatherAUT2021, 17