Package ‘restlos’

August 11, 2015

Type        Package
Title       Robust Estimation of Location and Scatter
Version     0.2-2
Date        2015-08-09
Author      Steffen Liebscher and Thomas Kirschstein
Maintainer  Steffen Liebscher <steffen.liebscher@wiwi.uni-halle.de>
Description The restlos package provides algorithms for robust estimation of location (mean and mode) and scatter based on minimum spanning trees (pMST), self-organizing maps (Flood Algorithm), Delaunay triangulations (RDELA), and nested minimum volume convex sets (MVCH). The functions are also suitable for outlier detection.
Depends     R (>= 3.2.1)
Imports     som (>= 0.3-5), rgl (>= 0.95.1247), geometry (>= 0.3-5),
            igraph (>= 1.0.1), limSolve (>= 1.5.5.1)
License     GPL (>= 2)
LazyLoad     yes
NeedsCompilation no
Repository   CRAN
Date/Publication 2015-08-11 15:28:34

R topics documented:

restlos-package ........................................... 2
flood ....................................................... 3
halle ....................................................... 4
MVCH ....................................................... 4
plot.flood ................................................ 6
plot.pMST ................................................ 7
plot.rdel ................................................ 8
pMST ....................................................... 9
rdela ..................................................... 10

Index 12
Description

The restlos package provides algorithms for robust estimation of location (mean and mode) and scatter based on minimum spanning trees (pMST), self-organizing maps (Flood Algorithm), Delaunay triangulations (RDELA), and nested minimum volume convex sets (MVCH). The functions are also suitable for outlier detection.

Details

Package: restlos
Type: Package
Version: 0.2-2
Date: 2015-08-09
License: GPL (>= 2)
LazyLoad: yes

Author(s)

Steffen Liebscher and Thomas Kirschstein
Maintainer: Steffen Liebscher <steffen.liebscher@wiwi.uni-halle.de>

References


**The Flood Algorithm**

**Description**

The function determines a robust subsample utilizing self-organizing maps (SOM).

**Usage**

```
flood(data, Nx=10, Ny=10, rlen=2000)
```

**Arguments**

- `data`: At least a two-dimensional data matrix is required. Number of observations needs to be greater than number of dimensions.
- `Nx`: Size of the SOM-net in x direction. Default is 10.
- `Ny`: Size of the SOM-net in y direction. Default is 10.
- `rlen`: Number of iterations during SOM learn process. Default is 2000.

**Details**

The function first calls the `som` function within the `som`-package. The results are subsequently used to determine a robust subsample. Arguments `Nx`, `Ny`, and `rlen` are passed to `som`. These arguments should be selected depending on the size of the data set (number of observations/dimensions). The larger the data set the larger the net size and the number of iterations should be. Note: At the moment only rectangular and quadratic SOM nets are supported.

**Value**

- `som.results`: SOM results as delivered by `som`.
- `som.neigh`: A matrix showing for every neuron (first column) the index of the neighboring neurons (columns 2-5).
- `umatrix`: The U-matrix shows the U-value for every neuron.
- `winneuron`: Vector of length `n` giving the index of the nearest neuron (Euclidean distance).
- `lib`: List of all basins found. Index of neurons. Smallest subsample of size `(n+d+1)/2`.
- `lin`: List of all neighboring neurons per basin. Index of neurons. Smallest subsample of size `(n+d+1)/2`.
- `geb`: Number of associated data points per basin. Smallest subsample of size `(n+d+1)/2`.
- `l`: Internal value necessary for plotting.
- `fafh`: Data for plotting the flood area flood height curve.
- `fafh.lib`: Internal data necessary for plotting extented flooding.
- `fafh.drin`: Internal data necessary for plotting extented flooding.
- `drin`: Robust subsample of minimal size.
Author(s)
Steffen Liebscher <steffen.liebscher@wiwi.uni-halle.de>

References

Examples
```r
# flood(halle)
```

---

**halle**

*Halle data set*

---

**Description**
Artificial data set containing a total of 1600 observations in two dimensions (in three groups with 100, 1000, 500 obs. from top to bottom). Central cluster is quadratically transformed leading to a U-shaped main part of the data.

**Usage**
```r
halle
```

**Examples**
```r
# plot(halle)
```

---

**MVCH**

*The MVCH Algorithm*

---

**Description**
The function determines the multivariate mode by iteratively selecting minimum volume convex subsets.

**Usage**
```r
MVCH(data, ps=0.75, pf=0.2, k=1000, a.poi=2, del.poi=1)
```
**Arguments**

- **data**: At least a two-dimensional data matrix is required. Number of observations needs to be greater than number of dimensions.

- **ps**: A numeric value between 0 and 1. Fraction of points to be retained in each iteration. Default is set to 0.7. See Details for more information.

- **pf**: A numeric value between 0 and 1. Fraction of points determining the size of the final subset. Default is set to 0.2. See Details for more information.

- **k**: The maximum number of iterations. Default is set to 1000. See Details for more information.

- **a.poi**: An integer $a.poi \geq 1$. Number of points added when searching for minimum volume subsets. Default is set to 2. See Details for more information.

- **del.poi**: An integer $1 \leq del.poi < a.poi$. Number of points deleted when searching for minimum volume subsets. Default is set to 1. See Details for more information.

**Details**

The algorithm iteratively determines a sequence of subsets of certain size with minimum convex hull volume (i.e. minimum volume subsets) until a certain threshold is reached. In the first iteration a minimum volume subset of size $n_1 = \lfloor n \cdot ps \rfloor$ is sought. In the second iteration, out of the subset found in iteration 1, a subset of size $n_2 = \lfloor n_1 \cdot ps \rfloor$ is determined. The procedure continues until the threshold is reached: $\lceil n \cdot pf \rceil$ where $n$ is the number of observations in data. The mode is calculated as the arithmetic mean of the observations in the final subset. Hence, the combination of $ps$ and $pf$ determines the running time and robustness of the procedure. Highest robustness (in terms of maximum breakdown point) is achieved for $ps = \lfloor \frac{n + d + 1}{2} \rfloor$. Small values of $pf$ guarantee an accurate mode estimation also for asymmetric data sets but running times increase.

To find a minimum volume subset, in each iteration $in.subs$ atomic subsets (consisting of $d+1$ observations) are constructed. Each of these atomic subsets is iteratively expanded by adding the $a.poi$ closest points and deleting $del.poi$. All three values determine the accuracy of the subset identification (and, hence, the estimate) as well as the running time of the algorithm. Small values of $in.subs$ reduce running time. Choosing similar values for $a.poi$ and $del.poi$ increases running time and algorithm accuracy.

For more details on the algorithm see the reference.

**Value**

A list with following entries:

- **mode**: The mode estimate.

- **set**: The final subset used for mode calculation.

- **vol**: The convex hull volume of the final subset.

- **set.1**: The subset identified after the first iteration (outlier-free subset).

**Author(s)**

Thomas Kirschstein <thomas.kirschstein@wiwi.uni-halle.de>
References


Examples

```r
# maximum breakdown point estimation
# MVCH(halle, ps = floor((nrow(halle) + ncol(halle) + 1)/2), pf = 0.05)

# slower estimation
# MVCH(halle, ps = 0.75, pf = 0.05)

# quicker estimation
# MVCH(halle, ps = 0.25, pf = 0.05)
```

---

**plot.flood**  
*Plot function for objects of class* flood

**Description**

Function to plot the results obtained by function flood

**Usage**

```r
## S3 method for class 'flood'
plot(x,..., level = 0)
```

**Arguments**

- `x` Object of class flood.
- `level` Flood level. Numeric value between 0 and 1. Default is 0 (i.e. all plots are based on the smallest robust subsample).
- `...` Further graphical parameters.

**Details**

The resulting plots depend on the dimensionality of the data set. For \(d=2\) and \(d=3\) the data set and the superimposed SOM net are plotted. For \(d>3\) a Mahalanobis distance plot is generated instead. The U-landscape and the Flood-Area-Flood-Height-curve are always plotted.

**Note**

At the moment no additional graphical parameters can be passed.
Author(s)

Steffen Liebscher <steffen.liebscher@wiwi.uni-halle.de>

References


Examples

```r
# plot(flood(halle))
```

---

**Description**

Function to plot the results obtained by function `pmst`.

**Usage**

```r
## S3 method for class 'pmst'
plot(x, ...)```

**Arguments**

- `x` Object of class `pmst`
- `...` Further graphical parameters.

**Details**

The resulting plots display the LC- and the AL-plot to support the decision on the size of the robust subsample, see references. Moreover, if the data set has dimension 2 or 3, the data set is plotted with the chosen robust subset superimposed as red points.

Author(s)

Thomas Kirschstein <thomas.kirschstein@wiwi.uni-halle.de>

References


Examples

```r
# plot(pmST(halle))
```
plot.rdela  

Plot function for objects of class rdela

Description

Function to plot the results obtained by function rdela

Usage

```r
CC sS method for class 'rdela'
plot(x,...)
```

Arguments

- `x` Object of class rdela.
- `...` Further graphical parameters.

Details

The resulting plots depend on the dimensionality of the data set. For d=2 and d=3 the data set and the selected robust subsample are plotted. For d>3 a Mahalanobis distance plot is generated instead.

Note

At the moment no additional graphical parameters can be passed.

Author(s)

Steffen Liebscher <steffen.liebscher@wiwi.uni-halle.de>

References


Examples

```r
# plot(rdela(halle))
```
The pMST Algorithm

Description
The function determines a robust subsample and computes estimates of location and scatter on the subset.

Usage
pMST(data, N = floor((nrow(data) + ncol(data) + 1)/2), lmax = nrow(data) * 100)

Arguments
- data: data set to be analyzed, at least a 2-dimensional matrix whose number of rows (i.e. observations \( n \)) is greater than the number of columns (i.e. dimension \( d \)).
- N: Size of the (robust) subsample to be determined. Default is \((n+d+1)/2\).
- lmax: Numerical option: determines the maximal number of pruning steps, see deteils.

Details
The function uses the minimum.spanning.tree function from the igraph-package to determine the minimum spanning tree (MST) of the data. The resulting MST is iteratively pruned by deleting edges (starting with the longest edge in the MST) until a connected subset with sufficient size (\( N \)) remains. Based on the robust subsample, location and scatter are estimated.

Value
- loc: Location estimate based on the robust subsample.
- cov: Covariance estimate based on the robust subsample.
- sample: Index of the observations in the robust subsample.
- data: The input data set.

Author(s)
Thomas Kirschstein <thomas.kirschstein@wiwi.uni-halle.de>

References

The RDELA Algorithm

Description

The function determines a robust subsample utilizing the Delaunay triangulation.

Usage

\[
\text{rdela}(\text{data, N, rew=TRUE})
\]

Arguments

- **data**: At least a two-dimensional data matrix is required. Number of observations needs to be greater than the number of dimensions. No degenerated (i.e. collinear) data sets allowed.
- **N**: Size of the identified subsample. Default is \((n+d+1)/2\).
- **rew**: Logical. Specifies whether reweighting should be conducted (TRUE) or not (FALSE). Default is TRUE.

Details

The function first calls the \texttt{delaunayn} function within the \texttt{geometry}-package. The results are subsequently used to determine a robust subsample.

Value

- **data**: The input data set.
- **tri**: Vertices of all simplices of the Delaunay triangulation. Each row represents a simplex.
- **neigh**: Lists for every simplex the adjacent/neighboring simplices. Each list entry represents a simplex.
- **radii**: Circum-(hypersphere-)radius of each simplex.
- **center**: Center coordinates of all simplices.
- **LiB**: List of all basins found. Index of simplices. Smallest subsample of size \((n+d+1)/2\).
- **LiN**: List of all neighboring simplices per basin. Index of simplices. Smallest subsample of size \((n+d+1)/2\).
- **GeB**: Number of associated data points per basin. Smallest subsample of size \((n+d+1)/2\).
**rdela**

**drin** (Initial) Robust subsample of size N.

**raw.mean** Mean estimate based on (initial) robust subsample of size N.

**raw.cov** Covariance estimate based on (initial) robust subsample of size N.

**final** Final robust subsample after reweighting.

**mean** Mean estimate based on final robust subsample.

**cov** Covariance estimate based on final robust subsample.

**Author(s)**

Steffen Liebscher <steffen.liebscher@wiwi.uni-halle.de>

**References**


**Examples**

```r
gerda(halle)
```
Index

†Topic datasets
  halle, 4
†Topic robust
  flood, 3
  MVCH, 4
  plot.flood, 6
  plot.pMST, 7
  plot.rdelta, 8
  pMST, 9
  rdelta, 10
  restlos-package, 2

flood, 3
halle, 4
MVCH, 4
plot.flood, 6
plot.pMST, 7
plot.rdelta, 8
pMST, 9
rdela, 10
restlos-package, 2
restlos-package, 2