Package 'cordillera'

January 14, 2018

Title Calculation of the OPTICS Cordillera

Version 0.8-0

Date 2018-01-13

Author Thomas Rusch [aut, cre], Patrick Mair [ctb], Kurt Hornik [ctb]

Maintainer Thomas Rusch <thomas.rusch@wu.ac.at>

Description Functions for calculating the OPTICS Cordillera. The OPTICS Cordillera measures the amount of 'clusteredness' in a numeric data matrix within a distance-density based framework for a given minimum number of points comprising a cluster, as described in Rusch, Hornik, Mair (2017) <doi:10.1080/10618600.2017.1349664>. There is an R native version and a version that uses 'ELKI', with methods for printing, summarizing, and plotting the result. There also is an interface to the reference implementation of OPTICS in 'ELKI'.

Depends R (>= 3.1.2),

SystemRequirements ELKI (>=0.6.0 if used)

Imports dbscan, yesno

Suggests cluster, scatterplot3d, MASS

License GPL-2 | GPL-3

LazyData true

URL http://r-forge.r-project.org/projects/stops/

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-01-14 04:06:01 UTC

R topics documented:

`cordillera-package` *cordillera: The OPTICS Cordillera*

Description

A package for calculating the OPTICS Cordillera. The package contains various functions, methods and classes for calculating and plotting the OPTICS Cordillera and an interface to ELKI's OPTICS.

Details

The stops package provides these main functions:

- cordillera() ... OPTICS Cordillera using dbscan OPTICS implementation
- e_cordillera()... ... OPTICS Cordillera using ELKI's OPTICS implementation
- e_optics() ... An interface to ELKI's implementation of OPTICS

Methods: For most of the objects returned by the high-level functions S3 classes and methods for standard generics were implemented, including print, summary, plot.

References:

- Rusch, T., Hornik, K., & Mair, P. (2017) Assessing and quantifying clusteredness: The OPTICS Cordillera, Journal of Computational and Graphical Statistics. http://dx.doi.org/10.1080/10618600.2017.1349664

Authors: Thomas Rusch

Maintainer: Thomas Rusch

Examples

```
data(CAClimateIndicatorsCountyMedian)

res<-princomp(CAClimateIndicatorsCountyMedian[,3:52])
res
summary(res)

library(scatterplot3d)
scatterplot3d(res$scores[,1:3])

irisrep3d<-res$scores[,1:3]
```

```
irisrep2d<-res$scores[,1:2]


#OPTICS in dbscan version
library(dbscan)
ores<-optics(irisrep2d,minPts=15,eps=100)
plot(ores)
#OPTICS cordillera for the 2D representation
cres2d<-cordillera(irisrep2d,minpts=15)
cres2d
summary(cres2d)
plot(cres2d)

#OPTICS cordillera for the 3D representation
cres3d<-cordillera(irisrep3d,minpts=15)
cres3d
summary(cres3d)
plot(cres3d)


#OPTICS in ELKI version
ores<-e_optics(irisrep2d,minpts=10,epsilon=100)
ores
summary(ores)
plot(ores)
```

`CAClimateIndicatorsCountyMedian`
Climate Change Indicators of Californian Counties

Description

A dataset containing various observed and projected indicators of climate change related natural hazards for 58 Californian counties. The values are actually the medians of the distribution over all spatial measurement points. It is a compiled data set from three sources and aggregated them to the county level. The projected data were derived under two different scenarios (A2, the high emission scenario and B1, the moderate emission scenario). It further contains the county value of the California social vulnerability index.

Format

A data frame with 58 rows and 52 variables

Details

Overall there are 50 indicators of natural hazard, one indicator of social vulnerability and 1 identifier of the county which were:

- county The county name identifier

- vuln_CA the vulnerability index
- County average 95th percentile daily maximum temperature in Fahrenheit from May 1 to September 30 over the historical period (1971-2000) under the two climate scenarios A2 and B1. These are averaged values for 4 different climate models. The source was Table 7 of Cooley (2012). The variables are degFA2 and degFB1.
- Projected average number of days where the daily maximum temperature exceeds the high-heat threshold (see above) over periods 2010-2039, 2040-2069 and 2070-2099. Projections are based on the A2 and B1 scenarios and are averaged for four downscaled climate models. The source was Table 7 of Cooley (2012). The variables are heatS_FYY_TYY, where S is the scenario, FYY the "from year" and TYY the "to year", so heatB1_71_00 is the value of a county under B1 from 1971 to 2000.
- The percentage of a county's census block area vulnerable to unimpeded coastal flooding under baseline conditions (2000) and with a 1.4-meter (55-inch) sea-level rise (projected for 2100). The raw data were obtained from Heberger (2009) . From the census block areas we computed an area-weighted percentage for each county. The variables are flood_2000 for 2000 and flood_2100 projected for 2100.
- The median aggregated Community Climate System Model v. 3 (CCSM3) projected annual actual evapotranspiration for years 2000, 2049 and 2099 under scenarios A2 and B1 by county. The source of the raw data was California Energy Commission (2008). The variables are evapS_YYYY, where S is the scenario and YYYY the year (observed or projected), so evapB1_2069 is the projected median evaporation in that county under B1 in 2069.
- The median aggregated CCSM3 projected annual baseflow for years 2000, 2049 and 2099 under scenarios A2 and B1 by county. The source of the raw data was California Energy Commission (2008). The variables are basfS_YYYY, where S is the scenario and YYYY the year (observed or projected), so basfB1_2069 is the projected median annual baseflow in that county under B1 in 2069.
- The median aggregated Centre National de Recherches Meteorologiques (CNRM) projected annual wildfire risk (observing 1 or more fires in the next 30 years). For years 2020 and 2085 under scenarios A2 and B1 by county. The source of the raw data was California Energy Commission (2008). The variables are fireS_YYYY, where S is the scenario and YYYY the year (observed or projected), so fireB1_2069 is the projected median wildfire risk in that county under B1 in 2069.
- The median aggregated CCSM3 projected annual fractional moisture in the entire soil column for years 2000, 2049 and 2099 under scenarios A2 and B1 by county. The source of the raw data was California Energy Commission (2008). The variables are smclS_YYYY, where S is the scenario and YYYY the year (observed or projected), so smclB1_2069 is the projected median soil column moisture in that county under B1 in 2069.
- The median aggregated CCSM3 projected annual precipitation for years 2000, 2049 and 2099 under scenarios A2 and B1 by county. The source of the raw data was California Energy Commission (2008). The variables are prcpS_YYYY, where S is the scenario and YYYY the year (observed or projected), so prcpB1_2069 is the projected median precipitation in that county under B1 in 2069.

Source

- Cooley (2012) http://www.energy.ca.gov/2012publications/CEC-500-2012-013/CEC-500-2012-013.pdf

- Heberger (2009) http://pacinst.org/reports/sea_level_rise/files/Blk_fld.zip
- California Energy Comission (2008) http://cal-adapt.org/data/

`cordillera` *The OPTICS Cordillera*

Description

Calculates the OPTICS Cordillera as described in Rusch et al. (2017). Based on optics in dbscan package.

Usage

```
cordillera(X, q = 2, minpts = 2, epsilon, distmeth = "euclidean",
  dmax = NULL, rang, digits = 10, scale = 4, ...)
```

Arguments

`X`	numeric matrix or data frame representing coordinates of points, or a symmetric matrix of distance of points or an object of class dist. Passed to `optics`, see also there.
`q`	The norm used for the Cordillera. Defaults to 2.
`minpts`	The minimum number of points that must make up a cluster in OPTICS (corresponds to k in the paper). It is passed to `optics` where it is called minPts. Defaults to 2.
`epsilon`	The epsilon parameter for OPTICS (called epsilon_max in the paper). Defaults to 2 times the maximum distance between any two points.
`distmeth`	The distance to be computed if X is not a symmetric matrix or a dist object (otherwise ignored). Defaults to Euclidean distance.
`dmax`	The winsorization value for the highest allowed reachability. If used for comparisons this should be supplied. If no value is supplied, it is NULL (default), then dmax is taken from the data as minimum of epsilon or the largest reachability.
`rang`	A range of values for making up dmax. If supplied it overrules the dmax parameter and rang[2]-rang[1] is returned as dmax in the object. If no value is supplied rang is taken to be (0, dmax) taken from the data. Only use this when you know what you're doing, which would mean you're me (and even then we should be cautious).
`digits`	The precision to round the raw Cordillera and the norm factor. Defaults to 10.
`scale`	Should X be scaled if it is an asymmetric matrix or data frame? Can take values TRUE or FALSE or a numeric value. If TRUE or 1, standardisation is to mean=0 and sd=1. If 2, no centering is applied and scaling of each column is done with the root mean square of each column. If 3, no centering is applied and scaling of all columns is done as X/max(standard deviation(allcolumns)). If 4, no centering is applied and scaling of all columns is done as X/max(rmsq(allcolumns)). If FALSE, 0 or any other numeric value, no standardisation is applied. Defaults to 4.
`...`	Additional arguments to be passed to `optics`

Value

A list with the elements

- $raw... The raw cordillera
- $norm... The normalization constant
- $normfac... The normalization factor (the number of times that dmax is taken)
- $dmaxe... The effective maximum distance used for maximum structure (either dmax or epsilon or rang[2]-rang[1]).
- $normed... The normed cordillera (raw/norm)
- $optics... The optics object

Warning

It may happen that the (normed) cordillera cannot be calculated properly (e.g. division by zero, infinite raw cordillera, q value to high etc.). A warning will be printed and the normed Cordillera is either 0, 1 (if infinity is involved) or NA. In that case one needs to check one or more of the following: reachability values returned from optics, minpts, eps, the raw cordillera, dmax and the normalization factor normfac.

Examples

```
data(iris)
res<-princomp(iris[,1:4])
#2 dim goodness-of-clusteredness with clusters of at least 2 points
#With a matrix of points
cres2<-cordillera(res$scores[,1:2])
cres2
summary(cres2)
plot(cres2)

#with a dist object
dl0 <- dist(res$scores[,1:2],"maximum") #maximum distance
cres0<-cordillera(dl0)
cres0
summary(cres0)
plot(cres0)

#with any symmetric distance/dissimilarity matrix
dl1 <- cluster::daisy(res$scores[,1:2],"manhattan")
cres1<-cordillera(dl1)
cres1
summary(cres1)
plot(cres1)

#4 dim goodness-of-clusteredness with clusters of at least 20
#points for PCA
cres4<-cordillera(res$scores[,1:4],minpts=20,epsilon=13,scale=3)
#4 dim goodness-of-clusteredness with clusters of at least 20 points for original
#data
cres<-cordillera(iris[,1:4],minpts=20,epsilon=13,dmax=cres4$dmaxe,scale=3)
```

```
#There is more clusteredness for the original result
summary(cres4)
summary(cres)
plot(cres4) #cluster structure only a bit intelligible
plot(cres) #clearly two well separated clusters

################################################################################
# Example from Rusch et al. (2017) with original data, PCA and Sammon mapping #
################################################################################

#data preparation
data(CAClimateIndicatorsCountyMedian)
sovisel <- CAClimateIndicatorsCountyMedian[,-c(1,2,4,9)]
#normalize to [0,1]
sovisel <- apply(sovisel,2,function(x) (x-min(x))/(max(x)-min(x)))
rownames(sovisel)  <- CAClimateIndicatorsCountyMedian[,1]
dis <- dist(sovisel)

#hyper parameters
dmax=1.22
q=2
minpts=3

#original data directly
cdat <- cordillera(sovisel,distmeth="euclidean",minpts=minpts,epsilon=10,q=q,
                   scale=0)
#equivalently
#dis2=dist(sovisel)
#cdat2 <- cordillera(dis2,minpts=minpts,epsilon=10,q=q,scale=FALSE)

#PCA in 2-dim
pca1 <- princomp(sovisel)
pcas <- scale(pca1$scores[,1:2])
cpca <- cordillera(pcas,minpts=minpts,epsilon=10,q=q,dmax=dmax,scale=FALSE)

#Sammon mapping in 2-dim
sam <- MASS::sammon(dis)
samp <- scale(sam$points)
csam <- cordillera(samp,epsilon=10,minpts=minpts,q=q,dmax=dmax,scale=FALSE)

#results
cdat
cpca
csam

par(mfrow=c(3,1))
plot(cdat)
plot(cpca)
plot(csam)
par(mfrow=c(1,1))
```

`e_cordillera` *Calculates the OPTICS Cordillera with the OPTICS implementation of 'ELKI'*

Description

Calculates the OPTICS cordillera as described in Rusch et al. (2017). Needs 'ELKI' >=0.6.0 - only tested with the Ubuntu binaries. This is an old implementation of the OPTICS Cordillera that relied on an external OPTICS implementation; since there is now an R package with an optics function the code has been re-factored. Only works with data matrices and Euclidean distance - `cordillera` is more general.

Usage

```
e_cordillera(confs, q = 1, minpts = 2, epsilon, dmax = NULL, rang,
  digits = 10, path = tempdir(), plot = FALSE, ylim, scale = 1, ...)
```

Arguments

`confs`	numeric matrix or data frame.
`q`	the norm of the OPTICS Cordillera. Defaults to 1.
`minpts`	the minpts argument to `elki`. Defaults to 2.
`epsilon`	The epsilon parameter for OPTICS. Defaults to 2 times the range of x.
`dmax`	The winsorization value for the highest allowed reachability. If used for comparisons this should be supplied. If no value is supplied, it is NULL (default), then dmax is taken from the data as minimum of epsilon or the largest reachability.
`rang`	(old parameter) A range of values for making up dmax. If supplied it overrules the dmax parameter and rang[2]-rang[1] is returned as dmax in the object. If no value is supplied rang is taken to be (0, dmax) taken from the data.
`digits`	round the raw OPTICS cordillera and the norm factor to these digits. Defaults to 10.
`path`	the path for storing the temporary files I/O files for optics. Defaults to tempdir(). In any other case it prompts the user for confirmation.
`plot`	plot the reachability and the raw OPTICS Cordillera
`ylim`	The borders for the OPTICS Cordillera plot
`scale`	Should the confs be scaled and/or centered? 0 does nothing, 1 does both, 2 only scales with the root mean square.
`...`	Additional arguments to be passed to optics

Value

A list with the elements

- $raw... The raw cordillera
- $norm... The normalization constant
- $normfac... The normalization factor (the number of times that dmax is taken)
- $dmax... The maximum distance used for maximum structure
- $normed... The normed cordillera (raw/norm)
- $optics... The optics object

Warning

It may happen that the (normed) cordillera cannot be calculated properly (e.g. division by zero, infinite raw cordillera, q value to high etc.). A warning will be printed and the normed cordillera is either 0, 1 (if infinity is involved) or NA. In that case one needs to check one or more of the following reachability values returned from optics, minpts, eps, the raw cordillera, dmax or the normalization factor.

`e_optics` *OPTICS in ELKI*

Description

A rudimentary I/O interface to OPTICS in 'ELKI' reroutes input from R to elki via the command line, runs elki.optics on an Input file (config.txt), sets up a temporary directory opticsout in path, reads in the contents of the `elki` output file clusterobjectorder.txt and deletes I/O files and directories. Returns the contents or clusterobjectorder.txt as a data frame. needs ELKI > 0.6.0 - Only tested with the Ubuntu trusty binaries very rudimentary as of yet. API of ELKI is not fixed. No plans for using this in the future.

Usage

```
e_optics(x, minpts = 2, epsilon, path = tempdir(), keep = FALSE)
```

Arguments

`x`	numeric matrix or data frame
`minpts`	the minpts argument to `elki`. Defaults to 2.
`epsilon`	The epsilon parameter for OPTICS. Defaults to 2 times the range of x.
`path`	the path for storing the temporary files I/O files. Defaults to tempdir().
`keep`	should the optics results from `elki` be stored in path. If TRUE, it will make a directory ./opticsout and a file config.txt

Value

a list with the contents of the `elki` output file as a data frame as element

- $clusterobjectorder, which is the clusterobjectorder file from `elki`. The first column is the OPTICS ordering (so the ID of the points in successive order), followed by the data in x, the next column lists the reachability followed by the predecessor of each point x[i,] in the ordering in the last column
- $eps
- $minpts

Examples

```
data(iris)
res<-e_optics(iris[,1:4],minpts=2,epsilon=100)
print(res)
summary(res)
plot(res,withlabels=TRUE)
```

---

oldcordilleraplot *Plot method for OPTICS Cordilleras. Deprecated.*

Description

Plots the reachability plot and adds the cordillera to it (as a line). In this plot the cordillera is proportional to the real value.

Usage

```
oldcordilleraplot(x, colbp = "lightgrey", coll = "black", liwd = 1.5,
  legend = FALSE, ylim, ...)
```

Arguments

`x`	an object of class cordillera
`colbp`	color of the barplot.
`coll`	color of the cordillera line
`liwd`	width of the cordillera line
`legend`	draw legend
`ylim`	ylim for the barplots
`...`	additional arguments passed to barplot or lines

plot.cordillera *Plot method for OPTICS Cordilleras*

Description

Plots the reachability plot and adds the cordillera to it (as a line). In this plot the cordillera is proportional to the real value.

Usage

```
## S3 method for class 'cordillera'
plot(x, colbp = "lightgrey", coll = "black",
  liwd = 1.5, legend = FALSE, ylim, ...)
```

Arguments

x	an object of class "cordillera"
colbp	color of the barplot.
coll	color of the cordillera line
liwd	width of the cordillera line
legend	draw legend
ylim	ylim for the barplots
...	additional arguments passed to barplot or lines

plot.opticse *Plot method for OPTICS results*

Description

Displays the reachability plot. Points with undefined/infinite minimum reachabilities are colored lighter by default.

Usage

```
## S3 method for class 'opticse'
plot(x, withlabels = FALSE, col = "grey55",
  colna = "grey80", border = graphics::par("bg"), names.arg, ...)
```

Arguments

x	an object of class optics
withlabels	flags whether point labels should be drawn. Defaults to TRUE.
col	the color of the bars of finite/defined reachabilities. Defaults to grey.
colna	the color of the bars for the points with infinite undefined reachabilities. Defaults to a lighter grey.
border	the color of the bar borders. Defaults to par("bg")
names.arg	... The arguments to be passed as names
...	additional arguments passed to barplot

---

print.cordillera *Print method for the OPTICS Cordillera*

Description

Prints the raw and normalized OPTICS Cordillera

Usage

```
## S3 method for class 'cordillera'
print(x, ...)
```

Arguments

x	an object of class optics
...	additional arguments passed to print

---

print.opticse *Print method for OPTICS results*

Description

Prints the object ordering and the reachabilities as directly read in from ELKI.

Usage

```
## S3 method for class 'opticse'
print(x, ...)
```

Arguments

x	an object of class optics
...	additional arguments passed to print

Value

a data frame with the observation order and the reachabilities (invisible)

---

print.summary.opticse *Print method for OPTICS summary*

Description

Displays summaries of the reachability plot. Currently its the five points summary of the reachabilities and a stem and leaf display. The latter should not be confused with the reachability plot. If you need the latter, use plot()

Usage

```
## S3 method for class 'summary.opticse'
print(x, fiven = TRUE, stemd = TRUE, ...)
```

Arguments

x	an object of class summary.optics
fiven	should the 5 point summary be printed. Default is TRUE.
stemd	should the stema dn leaf plot be printed. Default is TRUE.
...	additional arguments passed to stem

---

summary.opticse *Summary method for OPTICS results*

Description

Displays summaries of the reachability plot. Currently its the five points summary of the reachabilities and a stem and leaf display. The latter should not be confused with the reachability plot. If you need the latter, use plot().

Usage

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

Arguments

object	an object of class optics
...	additional arguments passed to summary.numeric

Value

an object of class summary.optics wit the reachabilities, the summary and minpts and epsilon parameters

Index