Package ‘pdc’

July 23, 2025

Type Package

Title Permutation Distribution Clustering

Version 1.0.3

Date 2015-09-28

Author Andreas M. Brandmaier

Maintainer Andreas M. Brandmaier <brandmaier@mpib-berlin.mpg.de>

Description Permutation Distribution Clustering is a clustering method for time series. Dissimilarity of time series is formalized as the divergence between their permutation distributions. The permutation distribution was proposed as measure of the complexity of a time series.

License GPL (>= 3)

Imports stats, utils, grDevices, graphics

Suggests plotrix, lattice

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-09-28 17:02:34

Contents

pdc-package *Permutation Distribution Clustering*

Description

Permutation Distribution Clustering (pdc) represents a complexity-based approach to clustering time series. Clustering comprises methods that recover similarities in a dataset and represent the findings in group structures. Important applications of clustering include the creation of taxonomies, the discovery of anomalies, or the the discovery of reliably different subgroups for differential analysis or treatment. A crucial parameter in clustering is the choice of the similarity measure between objects. Permutation Distribution Clustering finds similarity in time series based on differences in their permutation distribution as a proxy for differences in their complexity. The permutation distribution is obtained by counting the frequency of distinct order patterns in an m-embedding of the original time series. An embedding of dimension m allows for m! different order patterns. The choice of the embedding dimension crucially influences the clustering result. A small embedding dimension might lead to a permutation distribution with a low representational power, while a large embedding dimension quickly leads to a large permutation distribution that cannot reliably be estimated. With the Minimum Entropy Heuristic (MinE), the embedding dimension can automatically be chosen, thus making the algorithm a parameter-free clustering approach. For clustering timeseries, the similarity between two time-series is defined as the divergence between two permutation distributions. PDC is particularly apt for the analysis of psychophysiological time-series because it is efficient (the time complexity is linear in the time-series length), it is robust to drift, time-series of differing length can be compared, and it is invariant to differences in mean and variance of the time-series (choosing a normalization is not essential).

Details

The main function of the package is `pdclust`, which performs a hierarchical clustering of a set of time series based on differences in their permutation distributions. Other clustering or dimensionality-reduction methods can easily be employed by directly accessing the distance matrix based on the permutation distribution via `pdcDist`. A heuristic for choosing the embedding dimension is provided via `entropyHeuristic`. For clustering shapes, shape signatures can be traced from images with `traceImage`. Example data sets for shape clustering are `star.shapes` and `complex.shapes`.

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

See Also

`pdc` `pdcDist` `codebook`

Examples

```
# generate 5 ARMA time series for the first group
grp1 <- replicate(5, arima.sim(n = 500, list(ar = c(0.8897, -0.4858),
ma = c(-0.2279, 0.2488)),
           sd = sqrt(0.1796)) )

# generate 5 ARMA time series for the second group
grp2 <- replicate(5, arima.sim(n = 500, list(ar = c(-0.71, 0.18),
ma = c(0.92, 0.14)),
           sd = sqrt(0.291)) )

# combine groups into a single dataset
X <- cbind(grp1,grp2)

# run clustering and color original groups each in red and blue
clustering <- pdclust(X,3)
plot(clustering, cols=c(rep("red",5),rep("blue",5)))
```

---

codebook *Codebook*

Description

A codebook contains the permutation distribution of a time series.

Usage

```
codebook(x, m = 3, t = 1, use.fast=T, normalized = T)
```

Arguments

`x`	a vector ot a time series
`m`	The embedding dimension.
`t`	Time-delay of the embedding.
`use.fast`	Use a fast C-implementation if possible.
`normalized`	Normalize codebook such that it is a probability distribution.

Details

The length of a codebook is the factorial of the embedding dimension. The elements of the codebook represent relative frequencies of codewords of size m.

Value

Returns a vector of relative frequencies.

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

See Also

pdclust

Examples

```
# calculate codebook from sine-wave
cb <- codebook(c(sin(1:100)),m=3)

# plot the permutation distribution
barplot(cb,xlab="Permutation Distribution")
```

---

complex.shapes *Shape Signatures of Fish, Glasses, and Bottles*

Description

This data set provides exemplary shape signatures of each five fish, bottles, and pairs of glasses. The planar shapes were derived from original artwork obtained from openclipart.org. Column names encode the type of image.

Usage

```
data(complex.shapes)

data(complex.shapes.raw)
```

Format

`complex.shapes` is a matrix containing 100 rows and 15 columns obtained from `trace.image`. `complex.shapes.raw` contains a list containing 15 three-dimensional matrices and an array of corresponding labels.

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

`distance` *Codebook Dissimliarities*

Description

Functions to calculate distances/dissimilarities between codebooks.

Usage

```
hellingerDistance(x,y)
squaredHellingerDistance(x,y)
symmetricAlphaDivergence(x,y)
```

Arguments

`x`	a codebook
`y`	a codebook

Details

Note: The symmetric alpha-divergence is proportional to the Squared Hellinger distance, and is the default divergence between codebooks.

Value

Returns a numeric dissimilarity between two codebooks.

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

See Also

codebook

Examples

```
x <- codebook(c(sin(1:100)),m=3)
y <- codebook(c(sin(1:100*0.1)),m=3)
hellingerDistance(x,y)
```

entropyHeuristic *Minimum Entropy Heuristic (MinE)*

Description

The information content of a permutation distribution depends crucially on the choice of the embedding dimension. Too small embedding dimensions narrow the representational power of the distribution, too large embedding dimensions dilute the estimation of the distribution. The Minimum Entropy Heuristic (MinE) automatically chooses an embedding dimension with an optimal representational entropy as proxy for representational power.

Usage

```
entropyHeuristic(X, m.min=3, m.max=7, t.min = 1, t.max = 1)

## S3 method for class 'mine'
print(x, ...)
## S3 method for class 'mine'
summary(object, ...)
## S3 method for class 'mine'
plot(x, normalize = FALSE, type = "image", mark.optimum = TRUE,
                 col = heat.colors(12), ...)
```

Arguments

`X`	A matrix representing a set of time series. Columns are time series and rows represent time points.
`x`	An entropy heuristic object of type `mine`.
`object`	An entropy heuristic object of type `mine`.
`m.min`	Minimum embedding dimension
`m.max`	Maximum embedding dimension
`t.min`	Minimum time-delay
`t.max`	Maximum time-delay
`...`	Further arguments for the generic print, summary, and plot method.
`normalize`	Normalize values to range [0;1].
`type`	Either 'image' or 'contour'. Specifies the plot type.
`mark.optimum`	Mark the optimal embedding dimension and/or time-delay.
`col`	A color map to represent entropy values on.

Details

For a range of embedding dimensions, the average entropy of the dataset is calculated. The embedding dimension with the lowest entropy is chosen. `print` and `plot` is available for result objects.

The plot of a heuristic object shows the entropy values depending on a range of embedding dimensions and time-delays. If only embedding dimension or only time-delay is varied, a line plot is show to indicate the parameter yielding minimum entropy. Otherwise, an image plot is shown that indicates minimum entropy depending on both parameters.

Value

A list is returned with the following elements:

`m`	The chosen embedding size.
`entropy.values`	A vector with average entropy values corresponding to each entry in `entropy.ms`
`entropy.ms`	A vector of the embedding dimensions that were searched for the optimal embedding.

Author(s)

Andreas Brandmaier `<brandmaier@mpib-berlin.mpg.de>`

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

Brandmaier, A. M. (2012). *Permutation Distribution Clustering and Structural Equation Model Trees*. Doctoral dissertation. Saarland University, Saarbruecken, Germany.

See Also

`pdclust`

Examples

```
# (1)
#
# create a sine-wave with added noise
# and display a plot showing the average permutation entropy
# depending on varying choices of the embedding size
# (by default time-delay is not searched over)

heuristic <-  entropyHeuristic( sin(1:100)+rnorm(100,0,1) )
plot(heuristic)


# (2)
#
# calculate both optimal embedding dimension and time-delay
#
```

```
heuristic <-  entropyHeuristic( sin(1:100)+rnorm(100,0,1), t.min=1, t.max=6 )
plot(heuristic)
```

---

loo1nn *Leave-one-out One-nearest-neighbor Evaluation*

Description

Evaluates a clustering distance matrix within a supervised learning scheme: leave-one-out one-neares-neighbor cross-validation. This yields a rough estimate of the suitabilty of a distance function for discriminating between classes if ground-truth is known.

Usage

```
loo1nn(x, y)
```

Arguments

x	A `pdclust` object.
y	A vector of the true class labels.

Value

Returns a percentage-correct estimate.

Author(s)

Andreas Brandmaier `<brandmaier@mpib-berlin.mpg.de>`

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

See Also

`pdc.dist` `pdclust`

mdsPlot *Multidimensional Scaling Plot*

Description

Plots a two-dimensional projection to the principal coordinates of all observations. Clusters are shown as polygonal convex hulls of their members.

Usage

```
mdsPlot (X, labels = NULL, col = "gray")
```

Arguments

X	A pdclust object.
labels	Optional. A vector of labels for the observations. If NULL, column names of the dataset are used.
col	A vector of colors for polygon shading.

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

See Also

pdclust

Examples

```
data("complex.shapes")
truth <- c(rep("fish",5),rep("bottle",4),rep("glasses",5))
clust <- pdclust(complex.shapes, t=5)
mdsPlot(clust, truth, col=c("lightblue","lightgreen","lightgray"))
```

pdcDist *Permutation Distribution Clustering Distance Matrix*

Description

This function computes and returns the distance matrix computed by the divergence between permutation distributions of time series.

Usage

```
pdcDist(X, m = NULL, t = NULL, divergence = symmetricAlphaDivergence)
```

Arguments

X	A matrix representing a set of time series. Columns are time series and rows represent time points.
m	Embedding dimension for calculating the permutation distributions. Reasonable values range usually somewhere between 2 and 10. If no embedding dimension is chosen, the MinE heuristic is used to determine the embedding dimension automatically.
t	Time-delay of the embedding
divergence	Divergence measure between discrete distributions. Default is the symmetric alpha divergence.

Details

A valid divergence is always non-negative.

Value

Returns the dissimilarity between two codebooks as floating point number (larger or equal than zero).

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

See Also

pdclust

hclust kmeans

Examples

```
# create a set of time series consisting
# of sine waves with different degrees of added noise
# and two white noise time series
X <- cbind(
sin(1:500)+rnorm(500,0,.1),
sin(1:500)+rnorm(500,0,.2),
sin(1:500)+rnorm(500,0,.3),
sin(1:500)+rnorm(500,0,.4),
rnorm(500,0,1),
rnorm(500,0,1)
)

# calculate the distance matrix
D <- pdcDist(X,3)

# and plot with lattice package, you will
# be able to spot two clusters: a noise cluster
# and a sine wave cluster
require("lattice")
levelplot(as.matrix(D), col.regions=grey.colors(100,start=0.9, end=0.3))
```

`pdclust` *Permutation Distribution Clustering*

Description

Hierarchical cluster analysis for time series. Similarity of time series is based on the similarity of their permutation distributions.

Usage

```
pdclust(X, m = NULL, t = NULL, divergence =
                  symmetricAlphaDivergence, clustering.method =
                  "complete")


## S3 method for class 'pdclust'
plot(x, labels=NULL, type="rectangle", cols="black",
timeseries.as.labels = T, p.values=F, ...)


## S3 method for class 'pdclust'
str(object, ...)

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

Arguments

`X` In the univariate case: A matrix representing a set of time series. Columns represent different time series and rows represent time. In the multivariate case: A three-dimensional matrix with the first dimension representing time, second dimension representing multivariate time series, and the third dimension representing variables.

`m` Embedding dimension for calculating the permutation distributions. Reasonable values range somewhere between 2 and 10. If no embedding dimension is chosen, the MinE heuristic is used to determine the embedding dimension automatically.

`t` Time-delay of the embedding.

`divergence` Divergence measure between discrete distributions. Default is the symmetric alpha divergence.

`clustering.method` Hierarchical clustering linkage method. One out of c("complete","average","single").

For plotting:

`x` A `pdclust` object

`labels` Optionally provide a vector of labels for the time series here.

`type` One of c("triangle","rectangle") to choose the dendrogram style.

`cols` Specify line color either as string or as vector of strings

`timeseries.as.labels` If `FALSE`, a vertical dendrogram is plotted using hclust. If `TRUE`, a horizontal dendrogram is plotted with time series plots as labels.

`p.values` Annotation of the cluster hierarchy with p values

`...` Further graphical arguments.

For string representation:

`object` A `pdclust` object

Details

The function `pdclust` is the central function for clustering time-series in the package `pdc`. It allows clustering of univariate and multivariate time-series. If time-series have different length, the shorter time-series can be padded with `NA`s to bring them to columns of the same length in an array or a matrix. Multivariate time-series can also be handled by `pdclust`. Therefore, the data must be transformed into a three-dimensional matrix with the dimenions representing (1) time, (2) entities, and (3) variables/channels.

Value

Calls to `pdclust` return a `pdclust` object. There are `print`, `str` and `plot` methods for `pdclust` objects.

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.
Brandmaier, A. M. (2012). *Permutation Distribution Clustering and Structural Equation Model Trees*. Doctoral dissertation. Saarland University, Saarbruecken, Germany.

See Also

pdcDist entropyHeuristic symmetricAlphaDivergence

Examples

```
# generate 5 ARMA time series for the first group
grp1 <- replicate(5, arima.sim(n = 500, list(ar = c(0.8897, -0.4858),
ma = c(-0.2279, 0.2488)),
           sd = sqrt(0.1796)) )

# generate 5 ARMA time series for the second group
grp2 <- replicate(5, arima.sim(n = 500, list(ar = c(-0.71, 0.18),
ma = c(0.92, 0.14)),
           sd = sqrt(0.291)) )

# combine groups into a single dataset
X <- cbind(grp1,grp2)

# run clustering and color original groups each in red and blue
clustering <- pdclust(X)
plot(clustering, cols=c(rep("red",5),rep("blue",5)))
```

---

rasterPlot *Dendrogram Plot with Images*

Description

Plots a horizontal dendrogram with images as leafs

Usage

```
rasterPlot(cl, raw, monochrome=FALSE, aspect, ...)
```

Arguments

`cl`	A `pdclust` object.
`raw`	List of RGB images in matrix format.
`monochrome`	Convert image to b/w representation.
`aspect`	Aspect ratio.
`...`	Further graphical arguments.

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

See Also

`pdclust`

Examples

```
data("complex.shapes")
data("complex.shapes.raw")
clust <- pdclust(complex.shapes, t=5)
rasterPlot(clust, complex.shapes.raw$images)
```

`star.shapes` *Shape Signatures of Stars*

Description

This data set provides exemplary shape signatures of Column names are coded 'X-Y-Z' with X indicating the number of points (4 = 4-point-star or 5 = 5-point-star), Y indicating the size (0=100%, 1=150%), Z indicating rotation (0=0 degree, 1=45 degree).

Usage

```
data(star.shapes)

data(star.shapes.raw)
```

Format

`star.shapes` is a matrix containing 100 rows and 8 columns obtained from `traceImage`. `star.shapes.raw` contains a list of three-dimensional matrices containing the raw images and an array of corresponding labels.

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

`traceImage` *Shape Tracing of an Image*

Description

Trace the shape of an image to create a shape signature

Usage

```
traceImage(img, resolution)

convertImage(imgrgb, threshold = 0.5)
```

Arguments

`img`	a matrix of size width x height representing a black/white image with 0=white and 1=black.
`resolution`	The angular resolution of the trace, i.e., the length of the resulting shape signature.
`imgrgb`	a matrix of size width x height x channels representing an RGB image.
`threshold`	The average intensity value that serves as boundary to separate black and white pixels.

Details

Shape signatures of objects can be created by unrolling their contour around its centroid across time. The resulting time series represents distance-to-center of points on the contour versus radial angle.

In order to create signatures for RGB images, convert the image with `convert.image` to a black-and-white image using a `threshold` between 0 and 1.

Exemplary datasets containing shape signatures for shape clustering are provided in this package as `star.shapes` and `complex.shapes`.

Value

Returns a list containing `angles` and corresponding `distances` from center.

Author(s)

Andreas Brandmaier <brandmaier@mpib-berlin.mpg.de>

References

Brandmaier, A. M. (2015). pdc: An R Package for Complexity-Based Clustering of Time Series. *Journal of Statistical Software, 67(5)*, 1–23.

See Also

pdclust

Examples

```
# create a filled rectangle in a 20x20 image
img <- matrix(0, nrow=20,ncol=20)
img[5:15,5:15] <- 1

# create shape signature
signature <- traceImage(img)

# plot both original image and shape signature
par(mfrow=c(1,3))
#layout(matrix(c(1,2,2), 1, 3, byrow = TRUE))
image(img)
plot(signature$angle, signature$distance,type="l",xlab="angle",ylab="distance")

# reconstruct radial plot

require("plotrix")
radial.plot(traceImage(img,resolution=500)$distance,start=0,rp.type="r",radial.lim=c(0,10))
```

Index