Package ‘TSdist’

April 28, 2017

Type Package

Title Distance Measures for Time Series Data

Version 3.4

Encoding UTF-8

Date 2017-04-28

Depends R (>= 3.0.3), proxy

Imports zoo, xts, dtw, TSclust, KernSmooth, locpol, pdc, longitudinalData, cluster, methods, graphics, stats

Suggests testthat

Author Usue Mori, Alexander Mendiburu, J.A. Lozano

Maintainer Usue Mori <usue.mori@ehu.eus>

Description
A set of commonly used distance measures and some additional functions which, although initially not designed for this purpose, can be used to measure the dissimilarity between time series. These measures can be used to perform clustering, classification or other data mining tasks which require the definition of a distance measure between time series.

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-04-28 12:56:01 UTC

R topics documented:

`TSdist-package` *Distance Measures for Time Series in R.*

Description

A complete set of distance measures specifically designed to deal with time series.

Details

Package:	TSdist
Type:	Package
Version:	3.1
Date:	2015-07-14
License:	GPL (>=2)

This package provides a comprehensive set of time series distance measures published in the literature and some additional functions which, although initially not designed for this purpose, can be used to measure the dissimilarity between time series. These measures can be used to perform clustering, classification or other data mining tasks which require the definition of a distance measure between time series. Some of the measures are specifically implemented for this package while other are originally hosted in other R packages. The measures included are:

- Lp distances `LPDistance`
- Distance based on the cross-correlation `CCorDistance`
- Short Time Series distance (STS) `STSDistance`
- Dynamic Time Warping (DTW) `DTWDistance`
- LB_Keogh lower bound for the Dynamic Time Warping distance `LBKeoghDistance`
- Edit Distance for Real Sequences (EDR) `EDRDistance`
- Longest Common Subsequence distance for real sequences(LCSS) `LCSSDistance`
- Edit Distance based on Real Penalty (ERP) `ERPDistance`
- Distance based on the Fourier Discrete Transform `FourierDistance`
- TQuest distance `TquestDistance`
- Dissim distance `DissimDistance`
- Autocorrelation-based dissimilarity `ACFDistance`.
- Partial autocorrelation-based dissimilarity `PACFDistance`.
- Dissimilarity based on LPC cepstral coefficients `ARLPCCepsDistance`.
- Model-based dissimilarity proposed by Maharaj (1996, 2000) `ARMahDistance`.
- Model-based dissimilarity proposed by Piccolo (1990) `ARPicDistance`.
- Compression-based dissimilarity measure `CDMDistance`.
- Complexity-invariant distance measure `CIDDistance`.
- Dissimilarities based on Pearson's correlation `CorDistance`.
- Dissimilarity index which combines temporal correlation and raw value behaviors `CortDistance`.
- Dissimilarity based on wavelet feature extraction `WavDistance`.
- Integrated periodogram based dissimilarity `IntPerDistance`.
- Periodogram based dissimilarity `PerDistance`.
- Symbolic Aggregate Aproximation based dissimilarity `MindistSaxDistance`.
- Normalized compression based distance `NCDDistance`.
- Dissimilarity measure cased on nonparametric forecasts `PredDistance`.

- Dissimilarity based on the integrated squared difference between the log-spectra `SpecISDDistance`.
- General spectral dissimilarity measure using local-linear estimation of the log-spectra `SpecLLRDistance`.
- Permutation Distribution Distance `PDCDistance`.
- Frechet distance `FrechetDistance`.

All the measures are implemented in separate functions but can also be invoked by means of the wrapper function `TSDistances`. Moreover, this distance enables the use of time series objects of type `ts`, `zoo` and `xts`.

As an additional functionality of the package, pairwise distances between all the time series in a database can be easily computed by using the `dist` function from the **proxy** package or the `TSDatabaseDistances` function included in the **TSdist** package.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano. Maintainer: <usue.mori@ehu.es>

References

Esling, P., & Agon, C. (2012). *Time-series data mining.* ACM Computing Surveys, 45(1), 1-34.

Liao, T. W. (2005). *Clustering of time series data-a survey*. Pattern Recognition, 38(11), 1857-1874.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., & Keogh, E. (2012). *Experimental comparison of representation methods and distance measures for time series data.* Data Mining and Knowledge Discovery, 26(2), 275-309.

David Meyer and Christian Buchta (2013). proxy: Distance and Similarity Measures. R package version 0.4-10. http://CRAN.R-project.org/package=proxy

Examples

```
library(TSdist);
```

`ACFDistance` *Autocorrelation-based Dissimilarity*

Description

Computes the dissimilarity between a pair of numeric time series based on their estimated autocorrelation coefficients.

Usage

```
ACFDistance(x, y, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
...	Additional parameters for the function. See diss.ACF for more information.

Details

This is simply a wrapper for the diss.ACF function of package **TSclust**. As such, all the functionalities of the diss.ACF function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

Galeano, P., & Pella, D. (2000). Multivariate Analysis in Vector Time Series Pedro Galeano and Daniel Pella. Resenhas, the Journal of the Institute of Mathematics and Statistics of the University of Sao Paolo, 4, 383–403.

Lei, H., & Sun, B. (2007). A Study on the Dynamic Time Warping in Kernel Machines. In 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (pp. 839–845).

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see TSDistances. To calculate distance matrices of time series databases using this measure see TSDatabaseDistances.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the
# TSdist package.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)
```

```
# Calculate the autocorrelation based distance between the two series using
# the default parameters:

ACFDistance(example.series3, example.series4)
```

ARLPCCepsDistance *Dissimilarity Based on LPC Cepstral Coefficients*

Description

Computes the dissimilarity between two numeric time series in terms of their Linear Predictive Coding (LPC) ARIMA processes.

Usage

```
ARLPCCepsDistance(x, y, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
...	Additional parameters for the function. See diss.AR.LPC.CEPS for more information.

Details

This is simply a wrapper for the diss.AR.LPC.CEPS function of package **TSclust**. As such, all the functionalities of the diss.AR.LPC.CEPS function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using ts, zoo or xts objects see TSDistances. To calculate distance matrices of time series databases using this measure see TSDatabaseDistances.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the
# TSdist package obtained from an ARIMA(3,0,2) process.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the ar.lpc.ceps distance between the two series using
# the default parameters. In this case an AR model is automatically
# selected for each of the series:

ARLPCCepsDistance(example.series3, example.series4)

# Calculate the ar.lpc.ceps distance between the two series
# imposing the order of the ARIMA model of each series:

ARLPCCepsDistance(example.series3, example.series4, order.x=c(3,0,2),
order.y=c(3,0,2))
```

ARMahDistance — *Model-based Dissimilarity Proposed by Maharaj (1996, 2000)*

Description

Computes the model based dissimilarity proposed by Maharaj.

Usage

```
ARMahDistance(x, y, ...)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`...`	Additional parameters for the function. See `diss.AR.MAH` for more information.

Details

This is simply a wrapper for the `diss.AR.MAH` function of package **TSclust**. As such, all the functionalities of the `diss.AR.MAH` function are also available when using this function.

Value

statistic	The statistic of the homogeneity test.
p-value	The p-value issued by the homogeneity test.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the
# TSdist package obtained from an ARIMA(3,0,2) process.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the ar.mah distance between the two series using
# the default parameters.

ARMahDistance(example.series3, example.series4)

# The p-value is almost 1, which indicates that the two series come from the same
# ARMA process.
```

ARPicDistance *Model-based Dissimilarity Measure Proposed by Piccolo (1990)*

Description

Computes the model based dissimilarity proposed by Piccolo.

Usage

```
ARPicDistance(x, y, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
...	Additional parameters for the function. See diss.AR.PIC for more information.

Details

This is simply a wrapper for the diss.AR.PIC function of package **TSclust**. As such, all the functionalities of the diss.AR.PIC function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using ts, zoo or xts objects see TSDistances. To calculate distance matrices of time series databases using this measure see TSDatabaseDistances.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the
# TSdist package obtained from an ARIMA(3,0,2) process.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the Piccolo distance between the two series using
# the default parameters. In this case an AR model is automatically
# selected for each of the series:
```

```
ARPicDistance(example.series3, example.series4)

# Calculate the Piccolo distance between the two series
# imposing the order of the ARMA model of each series:

ARPicDistance(example.series3, example.series4, order.x=c(3,0,2),
order.y=c(3,0,2))
```

CCorDistance   *Cross-correlation based distance.*

Description

Computes the distance measure based on the cross-correlation between a pair of numeric time series.

Usage

```
CCorDistance(x, y, lag.max=(min(length(x), length(y))-1))
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`lag.max`	Positive integer that defines the maximum lag considered in the cross-correlation calculations (default=`min(length(x), length(y))-1`).

Details

The cross-correlation based distance between two numeric time series is calculated as follows:

$$D = \sqrt{((1 - CC(x, y, 0)^2) / \sum (1 - CC(x, y, k)^2))}$$

where $CC(x, y, k)$ is the cross-correlation between x and y at lag k.

The summatory in the denominator goes from 1 to `lag.max`. In view of this, the parameter must be a positive integer no larger than the length of the series.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Liao, T. W. (2005). *Clustering of time series data-a survey*. Pattern Recognition, 38(11), 1857-1874.

Pree, H., Herwig, B., Gruber, T., Sick, B., David, K., & Lukowicz, P. (2014). On general purpose time series similarity measures and their use as kernel functions in support vector machines. Information Sciences, 281, 478–495.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the
# TSdist package.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the cross-correlation based distance
# using the default lag.max.

CCorDistance(example.series3, example.series4)

# Calculate the cross-correlaion based distance
# with lag.max=50.

CCorDistance(example.series3, example.series4, lag.max=50)
```

CDMDistance *Compression-based Dissimilarity measure*

Description

Computes the dissimilarity between two numeric series based on their size after compression.

Usage

```
CDMDistance(x, y, ...)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`...`	Additional parameters for the function. See `diss.CDM` for more information.

Details

This is simply a wrapper for the `diss.CDM` function of package **TSclust**. As such, all the functionalities of the `diss.CDM` function are also available when using this function.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the compression based distance between the two series using
# the default parameters.

CDMDistance(example.series3, example.series4)
```

CIDDistance *Complexity-Invariant Distance Measure For Time Series*

Description

Computes the dissimilarity between two numeric series of the same length by calculating a correction of the Euclidean distance based on the complexity estimation of the series.

Usage

```
CIDDistance(x, y)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.

Details

This is simply a wrapper for the `diss.CID` function of package **TSclust**. As such, all the functionalities of the `diss.CID` function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)
```

```
# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the compression based distance between the two series using
# the default parameters.

CIDDistance(example.series1, example.series2)
```

CorDistance *Dissimilarities based on Pearson's correlation*

Description

Computes two different distance measure based on Pearson's correlation between a pair of numeric time series of the same length.

Usage

```
CorDistance(x, y, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
...	Additional parameters for the function. See diss.COR for more information.

Details

This is simply a wrapper for the diss.COR function of package **TSclust**. As such, all the functionalities of the diss.COR function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

Golay, X., Kollias, S., Stoll, G., Meier, D., Valavanis, A., & Boesiger, P. (1998). A new correlation-based fuzzy logic clustering algorithm for FMRI. Magnetic Resonance in Medicine, 40(2), 249–260.

See Also

To calculate this distance measure using ts, zoo or xts objects see TSDistances. To calculate distance matrices of time series databases using this measure see TSDatabaseDistances.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the first correlation based distance between the series.

CorDistance(example.series1, example.series2)

# Calculate the second correlation based distance between the series
# by specifying \eqn{beta}.

CorDistance(example.series1, example.series2, beta=2)
```

CortDistance — *Dissimilarity Index Combining Temporal Correlation and Raw Value Behaviors*

Description

Computes the dissimilarity between two numeric series of the same length by combining the dissimilarity between the raw values and the dissimilarity between the temporal correlation behavior of the series.

Usage

```
CortDistance(x, y, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
...	Additional parameters for the function. See diss.CORT for more information.

Details

This is simply a wrapper for the `diss.CORT` function of package **TSclust**. As such, all the functionalities of the `diss.CORT` function are also available when using this function.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

Chouakria, A. D., & Nagabhushan, P. N. (2007). Adaptive dissimilarity index for measuring time series proximity. Advances in Data Analysis and Classification, 1(1), 5–21. http://doi.org/10.1007/s11634-006-0004-6

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the first correlation based distance between the series using the default
# parameters.

CortDistance(example.series1, example.series2)
```

`DissimDistance` *The Dissim distance is calculated.*

Description

Computes the Dissim distance between a pair of numeric series.

Usage

```
DissimDistance(x, y, tx, ty)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`tx`	If not constant, a numeric vector that specifies the sampling index of series `x`.
`ty`	If not constant, a numeric vector that specifies the sampling index of series `y`.

Details

The Dissim distance is obtained by calculating the integral of the Euclidean distance between the two series. The series are assumed to be linear between sampling points.

The two series must start and end in the same interval but they may have different and non-constant sampling rates. These sampling rates must be positive and strictly increasing. For more information see reference below.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Frentzos, E., Gratsias, K., & Theodoridis, Y. (2007). *Index-based Most Similar Trajectory Search*. In Proceeding of the IEEE 23rd International Conference on Data Engineering (pp. 816-825).

Esling, P., & Agon, C. (2012). Time-series data mining. ACM Computing Surveys (CSUR), 45(1), 1–34.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
#The objects example.series1 and example.series2 are two
#numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

#For information on their generation and shape see help
#page of example.series.

help(example.series)

#Calculate the Dissim distance assuming even sampling:

DissimDistance(example.series1, example.series2)

#Calculate the Dissim distance assuming uneven sampling:

tx<-unique(c(seq(2, 175, 2), seq(7, 175, 7)))
tx <- tx[order(tx)]
ty <- tx
DissimDistance(example.series1, example.series2, tx, ty)
```

DTWDistance *Dynamic Time Warping distance.*

Description

Computes the Dynamic Time Warping distance between a pair of numeric time series.

Usage

```
DTWDistance(x, y, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
...	Additional parameters for the function. See dtw for more information.

Details

This is simply a wrapper for the dtw function of package **dtw**. As such, all the functionalities of the dtw function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Giorgino T (2009). *Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package.* Journal of Statistical Software, 31(7), pp. 1-24. URL:http://www.jstatsoft.org/v31/i07/

Cuturi, M. (2011). Fast Global Alignment Kernels. In Proceedings of the 28th International Conference on Machine Learning (pp. 929–936).

Gaidon, A., Harchaoui, Z., & Schmid, C. (2011). A time series kernel for action recognition. In BMVC 2011 - British Machine Vision Conference (pp. 63.1–63.11).

Marteau, P.-F., & Gibet, S. (2014). On Recursive Edit Distance Kernels With Applications To Time Series Classification. IEEE Transactions on Neural Networks and Learning Systems, PP(6), 1–13.

Lei, H., & Sun, B. (2007). A Study on the Dynamic Time Warping in Kernel Machines. In 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (pp. 839–845).

Pree, H., Herwig, B., Gruber, T., Sick, B., David, K., & Lukowicz, P. (2014). On general purpose time series similarity measures and their use as kernel functions in support vector machines. Information Sciences, 281, 478–495.

See Also

To calculate a lower bound of the DTW distance see `LBKeoghDistance`.

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the TSdist
# package


data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the basic DTW distance for two series of different length.

DTWDistance(example.series3, example.series4)

# Calculate the DTW distance for two series of different length
# with a sakoechiba window of size 30:
```

```
DTWDistance(example.series3, example.series4, window.type="sakoechiba", window.size=30)

# Calculate the DTW distance for two series of different length
# with an assymetric step pattern

DTWDistance(example.series3, example.series4, step.pattern=asymmetric)
```

EDRDistance *Edit Distance for Real Sequences (EDR).*

Description

Computes the Edit Distance for Real Sequences between a pair of numeric time series.

Usage

```
EDRDistance(x, y, epsilon, sigma)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
epsilon	A positive threshold value that defines the distance.
sigma	If desired, a Sakoe-Chiba windowing contraint can be added by specifying a positive integer representing the window size.

Details

The basic Edit Distance for Real Sequences between two numeric series is calculated. The idea is to count the number of edit operations (insert, delete, replace) that are necessary to transform one series into the other.

For that, if the Euclidean distance between two points x_i and y_i is smaller that `epsilon` they will be considered equal ($d = 0$) and if they are farther apart, they will be considered different ($d = 1$). As a last detail, this distance permits gaps or sequences of points that are not matched with any other point.

The length of series `x` and `y` may be different. Furthermore, if desired, a temporal constraint may be added to the EDR distance. In this package, only the most basic windowing function, introduced by H.Sakoe and S.Chiba (1978), is implemented. This function sets a band around the main diagonal of the distance matrix and avoids the matching of the points that are farther in time than a specified σ.

The size of the window must be a positive integer value. Furthermore, the following condition must be fulfilled:

$$|length(x) - length(y)| < sigma$$

Value

d The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Chen, L., Ozsu, M. T., & Oria, V. (2005). *Robust and Fast Similarity Search for Moving Object Trajectories*. In Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (pp. 491-502).

Chen, L., & Ng, R. (2004). On The Marriage of Lp-norms and Edit Distance. In Proceedings of the Thirtieth International Conference on Very Large Data Bases (pp. 792–803).

Cuturi, M. (2011). Fast Global Alignment Kernels. In Proceedings of the 28th International Conference on Machine Learning (pp. 929–936).

Gaidon, A., Harchaoui, Z., & Schmid, C. (2011). A time series kernel for action recognition. In BMVC 2011 - British Machine Vision Conference (pp. 63.1–63.11).

Marteau, P.-F., & Gibet, S. (2014). On Recursive Edit Distance Kernels With Applications To Time Series Classification. IEEE Transactions on Neural Networks and Learning Systems, PP(6), 1–13.

Lei, H., & Sun, B. (2007). A Study on the Dynamic Time Warping in Kernel Machines. In 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (pp. 839–845).

Pree, H., Herwig, B., Gruber, T., Sick, B., David, K., & Lukowicz, P. (2014). On general purpose time series similarity measures and their use as kernel functions in support vector machines. Information Sciences, 281, 478–495.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the TSdist
# package.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the EDR distance for two series of different length
```

```
# with no windowing constraint:

EDRDistance(example.series3, example.series4, epsilon=0.1)

# Calculate the EDR distance for two series of different length
# with a window of size 30:

EDRDistance(example.series3, example.series4, epsilon=0.1, sigma=30)
```

ERPDistance *Edit Distance with Real Penalty (ERP).*

Description

Computes the Edit Distance with Real Penalty between a pair of numeric time series.

Usage

```
ERPDistance(x, y, g, sigma)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
g	The reference value used to penalize gaps.
sigma	If desired, a Sakoe-Chiba windowing contraint can be added by specifying a positive integer representing the window size.

Details

The basic Edit Distance with Real Penalty between two numeric series is calculated. Unlike other edit based distances included in this package, this distance is a metric and fulfills the triangle inequality.

The idea is to search for the minimal path in a distance matrix that describes the mapping between the two series. This distance matrix is built by using the Euclidean distance. However, unlike DTW, this distance permits gaps or sequences of points that are not matched with any other point. These gaps will be penalized based on the distance of the unmatched points from a reference value g.

As with other edit based distances, the length of `x` and `y` may be different.

Furthermore, if desired, a temporal constraint may be added to the ERP distance. In this package, only the most basic windowing function, introduced by H.Sakoe and S.Chiba (1978), is implemented. This function sets a band around the main diagonal of the distance matrix and avoids the matching of the points that are farther in time than a specified σ.

The size of the window must be a positive integer value. Furthermore, the following condition must be fulfilled:

$$|length(x) - length(y)| < sigma$$

Value

`d` The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Chen, L., & Ng, R. (2004). *On The Marriage of Lp-norms and Edit Distance*. In Proceedings of the Thirtieth International Conference on Very Large Data Bases (pp. 792-803).

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
#The objects example.series3 and example.series4 are two
#numeric series of length 100 and 120 contained in the TSdist
#package.

data(example.series3)
data(example.series4)

#For information on their generation and shape see
#help page of example.series.

help(example.series)

#Calculate the ERP distance for two series of different length
#with no windowing constraint:

ERPDistance(example.series3, example.series4, g=0)

#Calculate the ERP distance for two series of different length
#with a window of size 30:

ERPDistance(example.series3, example.series4, g=0, sigma=30)
```

`EuclideanDistance` *Euclidean distance.*

Description

Computes the Euclidean distance between a pair of numeric vectors.

Usage

```
EuclideanDistance(x, y)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.

Details

The Euclidean distance is computed between the two numeric series using the following formula:

$$D = \sqrt{(x_i - y_i)^2)}$$

The two series must have the same length. This distance is calculated with the help of the `dist` function of the `proxy` package.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

David Meyer and Christian Buchta (2015). proxy: Distance and Similarity Measures. R package version 0.4-14. http://CRAN.R-project.org/package=proxy

See Also

This function can also be invoked by the wrapper function `LPDistance`.

Furthermore, to calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

# For information on their generation and shape see help
# page of example.series.

help(example.series)
```

```
# Compute the Euclidean distance between them:

EuclideanDistance(example.series1, example.series2)
```

`example.database` *Example databases.*

Description

Example database saved both as a numeric matrix and as a `zoo` object.

Usage

```
data(example.database);
data(zoo.database);
```

Format

`example.database` is saved in a numerical matrix.

`zoo.database` is saved as a `zoo` object with a given temporal index.

Details

`example.database` is a numerical matrix conformed by six ARMA(3,2) series of coefficients AR=(1, -0.24, 0.1) and MA=(1, 1.2) and length 100 that are situated in a row-wise format. They are generated from innovation vectors obtained randomly from a normal distribution of mean 0 and standard deviation 1, but by setting different random seeds.

`zoo.database` is a copy of `example.database` but saved in a `zoo` object with a specific time index. The series are set in a column-wise format.

Examples

```
data(example.database);
data(zoo.database);


## In example.database the series are set in a row-wise format.

plot(example.database)[1,]

## In zoo.database the series are set in a column-wise format.

plot(zoo.database)[,1]
```

`example.database2` *Example synthetic database with series belonging to different classes.*

Description

Example synthetic database with series belonging to 6 different classes.

Usage

```
data(example.database2);
```

Format

`example.database2` a list conformed of the following two elements:

- data The 100 time series are stored in a numeric matrix, row-wise.
- classes A numerical vector of length 100 that takes values in {1,2,3,4,5,6}. Each element in the vector represents the class of one of the series.

Details

`example.database2` is a database conformed of 100 series of length 100 obtained from 6 different classes. Each class is represented by the following function:

The class to which each series belongs is given in the `classes` vector.

- Class 1: random function
$$f1(t) = 80 + r(t) + n(t)$$
- Class 2: periodic function
$$f2(t) = 80 + 15\sin(\frac{2\pi t + sh}{T}) + n(t)$$
- Class 3: increasing linear trend
$$f3(t) = f_3(t) = 80 + 0.4t + n(t) + sh$$
- Class 4: decreasing linear trend
$$f4(t) = 80 - 0.4t + n(t) + sh$$
- Class 5: piecewise linear function which takes a value of $80 + n(t)$ for the first L/2+sh of the series and a value of $90 + n(t)$ for the rest of the points.
- Class 6: piecewise linear function which takes a value of $90 + n(t)$ for the first L/2+sh of the series and a value of $80 + n(t)$ for the rest of the points.
  $r(t)$ is a random value issued from a $N(0, 3)$ distribution, L is the length of the series, 100 in this case, and T is the period and is defined as a third of the length of the series. $n(t)$ is a random noise obtained from a $N(0, 2.8)$ distribution.. Finally, sh is an integer value that takes a random value between $(-7, 7)$ and shifts the series sh positions to the right or left, depending on the sign.

Examples

```
data(example.database2);

## The "data" element of the list contains the time series, set in a row-wise format.

plot(example.database2$data)[1,]

## The "classes" element in example.database2 contains the classes of the series:

example.database2$classes
```

`example.database3` *Example synthetic database with series belonging to different classes.*

Description

Example synthetic database with ARMA series belonging to 5 different classes.

Usage

```
data(example.database3);
```

Format

`example.database3` a list conformed of the following two elements:

- data The 50 time series are stored in a numeric matrix, row-wise.
- classes A numerical vector of length 50 that takes values in {1,2,3,4,5}. Each element in the vector represents the class of one of the series.

Details

`example.database3` is a database conformed of 50 series of length 100 obtained from 5 different classes. Each class is obtained from a different initializations of an ARMA(3,2) process of coefficients AR=(1,-0.24,0.1) and MA=(1,1.2).

Random noise is added to all the series by sampling values from a $N(0, 1.7)$ distribution. R is obtained from the following formula:

Finally, all the series in the database are shifted sh positions to the right or left, sh being a random integer value extracted from $-15, ..., 15$ in each case.

Examples

```
data(example.database3);

##  The "data" element of the list contains the time series, set in a row-wise format.

plot(example.database3$data)[1,]

##  The "classes" element in example.database3 contains the classes of the series:

example.database3$classes
```

---

example.series *Example series.*

Description

Example series saved as numeric vectors and as zoo objects.

Usage

```
data(example.series1);
data(example.series2);
data(example.series3);
data(example.series4);
data(zoo.series1);
data(zoo.series2);
```

Format

`example.series1`, `example.series2`, `example.series3` and `example.series4` are saved in numerical vectors.

`zoo.series1` and `zoo.series2` are saved as `zoo` objects with a given temporal index.

Details

`example.series1` and `example.series2` are generated based on the Two Patterns synthetic database introduced by Geurts (2002).

`example.series3` and `example.series4` are two ARMA(3,2) series of coefficients AR=(1, -0.24, 0.1) and MA=(1, 1.2) and length 100 and 120 respectively. They are generated from a pair of innovation vectors obtained randomly from a normal distribution of mean 0 and standard deviation 1, but by setting different random seeds.

`zoo.series1` and `zoo.series2` are copies of `example.series1` and `example.series2` but with a specific time index.

References

Geurts, P. (2002). Contributions to decision tree induction: bias/variance tradeoff and time series classification. University of Liege, Belgium.

Examples

```
data(example.series1);
data(example.series2);
data(example.series3);
data(example.series4);
data(zoo.series1);
data(zoo.series2);

## Plot series

plot(example.series1, type="l")
plot(example.series2, type="l")
plot(example.series3, type="l")
plot(example.series4, type="l")
plot(zoo.series1)
plot(zoo.series2)
```

`FourierDistance` *Fourier Coefficient based distance.*

Description

Computes the distance between a pair of numerical series based on their Discrete Fourier Transforms.

Usage

```
FourierDistance(x, y, n = (floor(length(x) / 2) + 1))
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
n	Positive integer that represents the number of Fourier Coefficients to consider. (default=(floor(length(x) / 2) + 1))

Details

The Euclidean distance between the first `n` Fourier coefficients of series `x` and `y` is computed. The series must have the same length. Furthermore, `n` should not be larger than the length of the series.

Value

d The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Agrawal, R., Faloutsos, C., & Swami, A. (1993). *Efficient similarity search in sequence databases*. In Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms (Vol. 5, pp. 69-84).

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

# For information on their generation and shape see help
# page of example.series.

help(example.series)

# Calculate the Fourier coefficient based  distance using
# the default number of coefficients:

FourierDistance(example.series1, example.series2)

# Calculate the Fourier coefficient based  distance using
# only the first 20 Fourier coefficients:

FourierDistance(example.series1, example.series2, n=20)
```

`FrechetDistance` *Frechet distance*

Description

Computes the Frechet distance between two numerical trajectories.

Usage

```
FrechetDistance(x, y, tx, ty, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
tx	If not constant, a numeric vector that specifies the sampling index of series x.
ty	If not constant, a numeric vector that specifies the sampling index of series y.
...	Additional parameters for the function. See distFrechet for more information.

Details

This is essentially a wrapper for the distFrechet function of package **longitudinalData**. As such, all the functionalities of the distFrechet function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Christophe Genolini (2014). longitudinalData: Longitudinal Data. R package version 2.2. http://CRAN.R-project.org/package=longitudinalData

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

Eiter, T., & Mannila, H. (1994). Computing Discrete Frechet Distance. Technical Report. Retrieved from http://www.kr.tuwien.ac.at/staff/eiter/et-archive/cdtr9464.pdf

See Also

To calculate this distance measure using ts, zoo or xts objects see TSDistances. To calculate distance matrices of time series databases using this measure see TSDatabaseDistances.

Examples

```
# The objects example.serie3 and example.series4 are two
# numeric series of length 100 and 120, respectively.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
```

```
# help page of example.series.

help(example.series)

# Calculate the distance based on wavelet feature extraction between the series.

## Not run: FrechetDistance(example.series3, example.series4)
```

`InfNormDistance` *The infinite norm distance.*

Description

Computes the infinite norm distance between a pair of numeric vectors.

Usage

```
InfNormDistance(x, y)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.

Details

The infinite norm distance is computed between the two numeric series using the following formula:

$$D = \max |x_i - y_i|$$

The two series must have the same length. This distance is calculated with the help of the `dist` function of the `proxy` package.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

David Meyer and Christian Buchta (2015). proxy: Distance and Similarity Measures. R package version 0.4-14. http://CRAN.R-project.org/package=proxy

See Also

This function can also be invoked by the wrapper function `LPDistance`.

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

# For information on their generation and shape see help
# page of example.series.

help(example.series)


# Compute the  infinite norm distance between them:

InfNormDistance(example.series1, example.series2)
```

`IntPerDistance` *Integrated Periodogram based dissimilarity*

Description

Calculates the dissimilarity between two numerical series of the same length based on the distance between their integrated periodograms.

Usage

```
IntPerDistance(x, y, ...)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`...`	Additional parameters for the function. See `diss.INT.PER` for more information.

Details

This is simply a wrapper for the `diss.INT.PER` function of package **TSclust**. As such, all the functionalities of the `diss.INT.PER` function are also available when using this function.

Value

d The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the ar.mah distance between the two series using
# the default parameters.

IntPerDistance(example.series1, example.series2)
```

KMedoids — *K medoids clustering for a time series database using the selected distance measure.*

Description

Given a specific distance measure and a time series database, this function provides the K-medoids clustering result. Furthermore, if the ground truth clustering is provided, and the associated F-value is also provided.

Usage

```
KMedoids(data,  k, ground.truth, distance, ...)
```

Arguments

`data`	Time series database saved in a numeric matrix, a list, an `mts` object, a `zoo` object or `xts` object.
`k`	Integer value which represents the number of clusters.
`ground.truth`	Numerical vector which indicates the ground truth clustering of the database.
`distance`	Distance measure to be used. It must be one of: `"euclidean"`, `"manhattan"`, `"minkowski"`, `"infnorm"`, `"ccor"`, `"sts"`, `"dtw"`, `"keogh_lb"`, `"edr"`, `"erp"`, `"lcss"`, `"fourier"`, `"tquest"`, `"dissimfull"`, `"dissimapprox"`, `"acf"`, `"pacf"`, `"ar.lpc.ceps"`, `"ar.mah"`, `"ar.mah.statistic"`, `"ar.mah.pvalue"`, `"ar.pic"`, `"cdm"`, `"cid"`, `"cor"`, `"cort"`, `"wav"`, `"int.per"`, `"per"`, `"mindist.sax"`, `"ncd"`, `"pred"`, `"spec.glk"`, `"spec.isd"`, `"spec.llr"`, `"pdc"`, `"frechet"`)
`...`	Additional parameters required by the chosen distance measure.

Details

This function is useful to evaluate the performance of different distance measures in the task of clustering time series.

Value

`clustering`	Numerical vector providing the clustering result for the database.
`F`	F-value corresponding to the clustering result.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

See Also

To calculate the distance matrices of time series databases the `TSDatabaseDistances` is used.

Examples

```
# The example.database3 synthetic database is loaded
data(example.database3)
tsdata <- example.database3[[1]]
groundt <- example.database3[[2]]

# Apply K-medoids clusterning for different distance measures

KMedoids(data=tsdata, ground.truth=groundt, k=5, "euclidean")
KMedoids(data=tsdata, ground.truth=groundt, k=5, "cid")
KMedoids(data=tsdata, ground.truth=groundt, k=5, "pdc")
```

`LBKeoghDistance` *LB_Keogh for DTW.*

Description

Computes the Keogh lower bound for the Dynamic Time Warping distance between a pair of numeric time series.

Usage

```
LBKeoghDistance(x, y, window.size)
```

Arguments

`x`	Numeric vector containing the first time series (query time series).
`y`	Numeric vector containing the second time series (reference time series).
`window.size`	Window size that defines the upper and lower envelopes.

Details

The lower bound introduced by Keogh and Ratanamahatana (2005) is calculated for the Dynamic Time Warping distance. Given `window.size`, the width of a Sakoe-Chiba band, an upper and lower envelope of the query time series is calculated in the following manner:

$$U[i] = max(x[i - window.size], x[i + window.size])$$

$$L[i] = min(x[i - window.size], x[i + window.size])$$

Based on this, the Keogh_LB distance is calculated as the Euclidean distance between the points in the reference time series (`y`) that fall outside both the lower and upper envelopes, and their nearest point of the corresponding envelope.

The series must have the same length. Furthermore, the width of the window should be even in order to assure a symmetric band around the diagonal and should not exceed the length of the series.

Value

`d`	The Keogh lower bound of the Dynamic Time Warping distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Keogh, E., & Ratanamahatana, C. A. (2004). Exact indexing of dynamic time warping. Knowledge and Information Systems, 7(3), 358-386.

Sakoe, H., & Chiba, S. (1978). *Dynamic programming algorithm optimization for spoken word recognition*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 26(1), 43-49.

Esling, P., & Agon, C. (2012). Time-series data mining. ACM Computing Surveys (CSUR), 45(1), 1–34.

See Also

To calculate the full DTW distance see `DTWDistance`.

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

# For information on their generation and shape see help
# page of example.series.

help(example.series)

# Calculate the LB_Keogh distance measure for these two series
# with a window of band of width 11:

LBKeoghDistance(example.series1, example.series2, window.size=11)
```

LCSSDistance — *Longest Common Subsequence distance for Real Sequences.*

Description

Computes the Longest Common Subsequence distance between a pair of numeric time series.

Usage

```
LCSSDistance(x, y, epsilon, sigma)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`epsilon`	A positive threshold value that defines the distance.
`sigma`	If desired, a Sakoe-Chiba windowing contraint can be added by specifying a positive integer representing the window size.

Details

The Longest Common Subsequence for two real sequences is computed.

For this purpose, the distances between the points of `x` and `y` are reduced to 0 or 1. If the Euclidean distance between two points x_i and y_j is smaller than `epsilon` they are considered equal and their distance is reduced to 0. In the opposite case, the distance between them is represented with a value of 1.

Once the distance matrix is defined in this manner, the maximum common subsequence is seeked. Of course, as in other Edit Based Distances, gaps or unmatched regions are permitted and they are penalized with a value proportional to their length.

Based on its definition, the length of series `x` and `y` may be different.

If desired, a temporal constraint may be added to the LCSS distance. In this package, only the most basic windowing function, introduced by H.Sakoe and S.Chiba (1978), is implemented. This function sets a band around the main diagonal of the distance matrix and avoids the matching of the points that are farther in time than a specified σ.

The size of the window must be a positive integer value. Furthermore, the following condition must be fulfilled:

$$|length(x) - length(y)| < sigma$$

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Vlachos, M., Kollios, G., & Gunopulos, D. (2002). *Discovering similar multidimensional trajectories*. In Proceedings 18th International Conference on Data Engineering (pp. 673-684). IEEE Comput. Soc. doi:10.1109/ICDE.2002.994784

Chen, L., & Ng, R. (2004). On The Marriage of Lp-norms and Edit Distance. In Proceedings of the Thirtieth International Conference on Very Large Data Bases (pp. 792–803).

Cuturi, M. (2011). Fast Global Alignment Kernels. In Proceedings of the 28th International Conference on Machine Learning (pp. 929–936).

Gaidon, A., Harchaoui, Z., & Schmid, C. (2011). A time series kernel for action recognition. In BMVC 2011 - British Machine Vision Conference (pp. 63.1–63.11).

Marteau, P.-F., & Gibet, S. (2014). On Recursive Edit Distance Kernels With Applications To Time Series Classification. IEEE Transactions on Neural Networks and Learning Systems, PP(6), 1–13.

Lei, H., & Sun, B. (2007). A Study on the Dynamic Time Warping in Kernel Machines. In 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (pp. 839–845).

Pree, H., Herwig, B., Gruber, T., Sick, B., David, K., & Lukowicz, P. (2014). On general purpose time series similarity measures and their use as kernel functions in support vector machines. Information Sciences, 281, 478–495.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the TSdist
# package.


data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the LCSS distance for two series of different length
# with no windowing constraint:

LCSSDistance(example.series3, example.series4, epsilon=0.1)

# Calculate the LCSS distance for two series of different length
# with a window of size 30:

LCSSDistance(example.series3, example.series4, epsilon=0.1, sigma=30)
```

LPDistance *Lp distances.*

Description

Computes the distance based on the chosen Lp norm between a pair of numeric vectors.

Usage

```
LPDistance(x, y, method="euclidean", ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
method	A value in "euclidean", "manhattan", "infnorm", "minkowski".
...	If method="minkowski" a positive integer value must be specified for p.

Details

The distances based on Lp norms are computed between two numeric vectors using the following formulas:

Euclidean distance: $\sqrt{(x_i - y_i)^2)}$

Manhattan distance: $\sum |x_i - y_i|$

Infinite norm distance: $\max |x_i - y_i|$

Minkowski distance: $\sqrt[p]{(x_i - y_i)^p)}$

The two series must have the same length. Furthermore, in the case of the Minkowski distance, p must be specified as a positive integer value.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

See Also

These distances are also implemeted in separate functions. For more information see EuclideanDistance, ManhattanDistance, MinkowskiDistance and InfNormDistance

To calculate this distance measure using ts, zoo or xts objects see TSDistances. To calculate distance matrices of time series databases using this measure see TSDatabaseDistances.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

# For information on their generation and shape see help
# page of example.series.
```

```
help(example.series)

# Compute the different Lp distances

# Euclidean distance

LPDistance(example.series1, example.series2, method="euclidean")

# Manhattan distance

LPDistance(example.series1, example.series2, method="manhattan")

# Infinite norm distance

LPDistance(example.series1, example.series2, method="infnorm")

# Minkowski distance with p=3.

LPDistance(example.series1, example.series2, method="minkowski", p=3)
```

`ManhattanDistance` *Manhattan distance.*

Description

Computes the Manhattan distance between a pair of numeric vectors.

Usage

```
ManhattanDistance(x, y)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.

Details

The Manhattan distance is computed between the two numeric series using the following formula:

$$D = \sum |x_i - y_i|$$

The two series must have the same length. This distance is calculated with the help of the `dist` function of the `proxy` package.

Value

d The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

David Meyer and Christian Buchta (2015). proxy: Distance and Similarity Measures. R package version 0.4-14. http://CRAN.R-project.org/package=proxy

See Also

This function can also be invoked by the wrapper function `LPDistance`.

Furthermore, to calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

# For information on their generation and shape see help
# page of example.series.

help(example.series)

# Compute the Manhattan distance between them:

ManhattanDistance(example.series1, example.series2)
```

`MindistSaxDistance` *Symbolic Aggregate Aproximation based dissimilarity*

Description

Calculates the dissimilarity between two numerical series based on the distance between their SAX representations.

Usage

```
MindistSaxDistance(x, y, w, ...)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`w`	The amount of equal sized windows that the series will be reduced to.
`...`	Additional parameters for the function. See `diss.MINDIST.SAX` for more information.

Details

This is simply a wrapper for the `diss.MINDIST.SAX` function of package **TSclust**. As such, all the functionalities of the `diss.MINDIST.SAX` function are also available when using this function.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 respectively.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the mindist.sax distance between the two series using
# 20 equal sized windows for each series. The rest of the parameters
# are left in their default mode.

MindistSaxDistance(example.series3, example.series4, w=20)
```

`MinkowskiDistance` *Minkowski distance.*

Description

Computes the Minkowski distance between two numeric vectors for a given p.

Usage

```
MinkowskiDistance(x, y, p)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`p`	A strictly positive integer value that defines the chosen L_p norm.

Details

The Minkowski distance is computed between the two numeric series using the following formula:

$$D = \sqrt[p]{(x_i - y_i)^p)}$$

The two series must have the same length and `p` must be a positive integer value. This distance is calculated with the help of the `dist` function of the `proxy` package.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

David Meyer and Christian Buchta (2015). proxy: Distance and Similarity Measures. R package version 0.4-14. http://CRAN.R-project.org/package=proxy

See Also

This function can also be invoked by the wrapper function `LPDistance`.

Furthermore, to calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

# For information on their generation and shape see help
# page of example.series.

help(example.series)


# Compute the Minkowski distance between them:

MinkowskiDistance(example.series1, example.series2, p=3)
```

`NCDDistance` *Normalized Compression based distance*

Description

Calculates a normalized distance between two numerical series based on their compressed sizes.

Usage

```
NCDDistance(x, y, ...)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`...`	Additional parameters for the function. See `diss.NCD` for more information.

Details

This is simply a wrapper for the `diss.NCD` function of package **TSclust**. As such, all the functionalities of the `diss.NCD` function are also available when using this function.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 respectively.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the normalized compression based distance between the two series
# using default parameter.

NCDDistance(example.series3, example.series4)
```

`OneNN` *1NN classification for a pair of train/test time series datasets.*

Description

Given a specific distance measure, this function provides the 1NN classification values and the associated error for a specific train/test pair of time series databases.

Usage

```
OneNN(train, trainc, test, testc, distance, ...)
```

Arguments

`train`	Time series database saved in a numeric matrix, a list, an `mts` object, a `zoo` object or `xts` object.
`trainc`	Numerical vector which indicates the class of each of the series in the training set.

`test`	Time series database saved in a numeric matrix, a list, an `mts` object, a `zoo` object or `xts` object.
`testc`	Numerical vector which indicates the class of each of the series in the testing set.
`distance`	Distance measure to be used. It must be one of: `"euclidean"`, `"manhattan"`, `"minkowski"`, `"infnorm"`, `"ccor"`, `"sts"`, `"dtw"`, `"keogh_lb"`, `"edr"`, `"erp"`, `"lcss"`, `"fourier"`, `"tquest"`, `"dissimfull"`, `"dissimapprox"`, `"acf"`, `"pacf"`, `"ar.lpc.ceps"`, `"ar.mah"`, `"ar.mah.statistic"`, `"ar.mah.pvalue"`, `"ar.pic"`, `"cdm"`, `"cid"`, `"cor"`, `"cort"`, `"wav"`, `"int.per"`, `"per"`, `"mindist.sax"`, `"ncd"`, `"pred"`, `"spec.glk"`, `"spec.isd"`, `"spec.llr"`, `"pdc"`, `"frechet"`)
`...`	Additional parameters required by the chosen distance measure.

Details

This function is useful to evaluate the performance of different distance measures in the task of classification of time series.

Value

`classes`	Numerical vector providing the predicted class values for the series in the test set.
`error`	Error obtained in the 1NN classification process.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

See Also

To calculate the distance matrices of time series databases the `TSDatabaseDistances` is used.

Examples

```
# The example.database2 synthetic database is loaded
data(example.database2)

# Create train/test by dividing the dataset 70%-30%
set.seed(100)
trainindex <- sample(1:100, 70, replace=FALSE)
train <- example.database2[[1]][trainindex, ]
test <- example.database2[[1]][-trainindex, ]
trainclass <- example.database2[[2]][trainindex]
testclass <- example.database2[[2]][-trainindex]

# Apply the 1NN classifier for different distance measures

OneNN(train, trainclass, test, testclass, "euclidean")
OneNN(train, trainclass, test, testclass, "pdc")
```

`PACFDistance` *Partial Autocorrelation-based Dissimilarity*

Description

Computes the dissimilarity between a pair of numeric time series based on their estimated partial autocorrelation coefficients.

Usage

```
PACFDistance(x, y, ...)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`...`	Additional parameters for the function. See `diss.PACF` for more information.

Details

This is simply a wrapper for the `diss.PACF` function of package **TSclust**. As such, all the functionalities of the `diss.PACF` function are also available when using this function.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the
# TSdist package.

data(example.series3)
data(example.series4)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the autocorrelation based distance between the two series using
# the default parameters:

PACFDistance(example.series3, example.series4)
```

PDCDistance *Permutation Distribution Distance*

Description

Calculates the permutation distribution distance between two numerical series of the same length.

Usage

```
PDCDistance(x, y, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
...	Additional parameters for the function. See pdcDist for more information.

Details

This is simply a wrapper for the pdcDist function of package **pdc**. As such, all the functionalities of the pdcDist function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Andreas M. Brandmaier (2015). pdc: An R package for Complexity-Based Clustering of Time Series. Journal of Statistical Software, Vol 67, Issue 5.

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the normalized compression based distance between the two series
# using the default parameters.

PDCDistance(example.series1, example.series2)
```

PerDistance — *Periodogram based dissimilarity*

Description

Calculates the dissimilarity between two numerical series of the same length based on the distance between their periodograms.

Usage

```
PerDistance(x, y, ...)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`...`	Additional parameters for the function. See `diss.PER` for more information.

Details

This is simply a wrapper for the `diss.PER` function of package **TSclust**. As such, all the functionalities of the `diss.PER` function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the ar.mah distance between the two series using
# the default parameters.

PerDistance(example.series1, example.series2)
```

PredDistance *Dissimilarity Measure Based on Nonparametric Forecasts*

Description

The dissimilarity of two numerical series of the same length is calculated based on the L1 distance between the kernel estimators of their forecast densities at a given time horizon.

Usage

```
PredDistance(x, y, h, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
h	Integer value representing the prediction horizon.
...	Additional parameters for the function. See diss.PRED for more information.

Details

This is simply a wrapper for the diss.PRED function of package **TSclust**. As such, all the functionalities of the diss.PRED function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using ts, zoo or xts objects see TSDistances. To calculate distance matrices of time series databases using this measure see TSDatabaseDistances.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the prediction based distance between the two series using
# the default parameters.
```

```
PredDistance(example.series1, example.series2)
```

`SpecGLKDistance` *Dissimilarity based on the Generalized Likelihood Ratio Test*

Description

The dissimilarity of two numerical series of the same length is calculated based on an adaptation of the generalized likelihood ratio test.

Usage

```
SpecGLKDistance(x, y, ...)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`...`	Additional parameters for the function. See `diss.PER` for more information.

Details

This function simply intends to be a wrapper for the `diss.SPEC.GLK` function of package **TSclust**. However, in the 1.2.3 version of the **TSclust** package we have found an error in the call to this function. As such, in this version, the more general `diss` function, designed for distance matrix calculations of time series databases, is used to calculate the spec.glk distance between two series. Once this bug is fixed in the original package, we will update our call procedure.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the ar.mah distance between the two series using
# the default parameters.

SpecGLKDistance(example.series1, example.series2)
```

SpecISDDistance — *Dissimilarity Based on the Integrated Squared Difference between the Log-Spectra*

Description

The dissimilarity of two numerical series of the same length is calculated based on the integrated squared difference between the non-parametric estimators of their log-spectra.

Usage

```
SpecISDDistance(x, y, ...)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`...`	Additional parameters for the function. See `diss.SPEC.ISD` for more information.

Details

This is simply a wrapper for the `diss.SPEC.ISD` function of package **TSclust**. As such, all the functionalities of the `diss.SPEC.ISD` function are also available when using this function.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the spec.isd distance between the two series using
# the default parameters.

SpecISDDistance(example.series1, example.series2)
```

`SpecLLRDistance` *General Spectral Dissimilarity Measure Using Local-Linear Estimation of the Log-Spectra*

Description

The dissimilarity of two numerical series of the same length is calculated based on the ratio between local linear estimations of the log-spectras.

Usage

```
SpecLLRDistance(x, y, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
...	Additional parameters for the function. See diss.SPEC.LLR for more information.

Details

This is simply a wrapper for the diss.SPEC.LLR function of package **TSclust**. As such, all the functionalities of the diss.SPEC.LLR function are also available when using this function.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the spec.isd distance between the two series using
# the default parameters.

SpecLLRDistance(example.series1, example.series2)
```

`STSDistance` *Short time series distance (STS).*

Description

Computes the Short Time Series Distance between a pair of numeric time series.

Usage

```
STSDistance(x, y, tx, ty)
```

Arguments

`x`	Numeric vector containing the first time series.
`y`	Numeric vector containing the second time series.
`tx`	If not constant, a numeric vector that specifies the sampling index of series `x`.
`ty`	If not constant, a numeric vector that specifies the sampling index of series `y`.

Details

The short time series distance between two series is designed specially for series with an equal but uneven sampling rate. However, it can also be used for time series with a constant sampling rate. It is calculated as follows:

$$STS = \sqrt{\sum}(((y_{k+1} - y_k)/(tx_{k+1} - tx_k) - (x_{k+1} - x_k)/(ty_{k+1} - ty_k))^2)$$

where N is the length of series x and y and the summatory goes from 1 to one minus the length of the series.

`tx` and `ty` must be positive and strictly increasing. Furthermore, the sampling rate in both indexes must be equal:

$$tx[k+1] - tx[k] = ty[k+1] - ty[k], \;\; for \;\; k = 0, ..., N-1$$

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Möller-Levet, C. S., Klawonn, F., Cho, K., & Wolkenhauer, O. (2003). *Fuzzy Clustering of Short Time-Series and Unevenly Distributed Sampling Points*. In Proceedings of the 5th International Symposium on Intelligent Data Analysis.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

# For information on their generation and shape see help
# page of example.series.

help(example.series)

# Calculate the STS distance assuming even sampling:

STSDistance(example.series1, example.series2)

# Calculate the STS distance providing an uneven sampling:

tx<-unique(c(seq(2, 175, 2), seq(7, 175, 7)))
tx <- tx[order(tx)]
ty <- tx
STSDistance(example.series1, example.series2, tx, ty)
```

`TquestDistance` *Tquest distance.*

Description

Computes the Tquest distance between a pair of numeric vectors.

Usage

```
TquestDistance(x, y, tx, ty, tau)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
tx	If not constant, temporal index of series x.
ty	If not constant, temporal index of series y.
tau	Parameter (threshold) used to define the threshold passing intervals.

Details

The TQuest distance represents the series based on a set of intervals that fulfill the following conditions:

1. All the values that the time series takes during these time intervals must be strictly above a user specified threshold `tau`.
2. They are the largest possible intervals that satisfy the previous condition.

The final distance between two series is defined in terms of the similarity between their threshold passing interval sets. For more information, see references.

Value

`d`	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Aßfalg, J., Kriegel, H., Kröger, P., Kunath, P., Pryakhin, A., & Renz, M. (2006). *Similarity Search on Time Series based on Threshold Queries*. In Proceedings of the 10th international conference on Advances in Database Technology (pp. 276-294).

Esling, P., & Agon, C. (2012). Time-series data mining. ACM Computing Surveys (CSUR), 45(1), 1–34.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see `TSDistances`. To calculate distance matrices of time series databases using this measure see `TSDatabaseDistances`.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100 contained in the TSdist package.

data(example.series1)
data(example.series2)

# For information on their generation and shape see help
# page of example.series.

help(example.series)


# Calculate the Tquest distance assuming even sampling:

TquestDistance(example.series1, example.series2, tau=2.5)
```

```
# The objects example.series3 and example.series4 are two
# numeric series of length 100 and 120 contained in the TSdist
# package.

data(example.series3)
data(example.series4)

# Calculate the Tquest distance for two series of different length:

TquestDistance(example.series3, example.series4, tau=2.5)
```

TSDatabaseDistances *TSdist distance matrix computation.*

Description

TSdist distance matrix computation for time series databases.

Usage

```
TSDatabaseDistances(X, Y=NULL, distance, ...)
```

Arguments

X	Time series database saved in a numeric matrix, a list, an mts object, a zoo object or xts object.
Y	Time series database saved in a numeric matrix, a list, an mts object, a zoo object or xts object. Should only be defined for calculation of distance matrices between two different databases so default value is NULL.
distance	Distance measure to be used. It must be one of: "euclidean", "manhattan", "minkowski", "infnorm", "ccor", "sts", "dtw", "keogh.lb", "edr", "erp", "lcss", "fourier", "tquest", "dissim", "acf", "pacf", "ar.lpc.ceps", "ar.mah", "ar.mah.statistic", "ar.mah.pvalue", "ar.pic", "cdm", "cid", "cor", "cort", "wav", "int.per", "per", "mindist.sax", "ncd", "pred", "spec.glk", "spec.isd", "spec.llr", "pdc", "frechet")
...	Additional parameters required by the chosen distance measure.

Details

The distance matrix of a time series database is calculated by providing the pair-wise distances between the series that conform it. x can be saved in a numeric matrix, a list or a mts, zoo or xts object. The following distance methods are supported:

- "euclidean": Euclidean distance. EuclideanDistance
- "manhattan": Manhattan distance. ManhattanDistance
- "minkowski": Minkowski distance. MinkowskiDistance

- "infnorm": Infinite norm distance. InfNormDistance
- "ccor": Distance based on the cross-correlation. CCorDistance
- "sts": Short time series distance. STSDistance
- "dtw": Dynamic Time Warping distance. DTWDistance. Uses the **dtw** package (see dtw).
- "lb.keogh": LB_Keogh lower bound for the Dynamic Time Warping distance. LBKeoghDistance
- "edr": Edit distance for real sequences. EDRDistance
- "erp": Edit distance with real penalty. ERPDistance
- "lcss": Longest Common Subsequence Matching. LCSSDistance
- "fourier": Distance based on the Fourier Discrete Transform. FourierDistance
- "tquest": TQuest distance. TquestDistance
- "dissim": Dissim distance. DissimDistance
- "acf": Autocorrelation-based dissimilarity ACFDistance. Uses the **TSclust** package (see diss.ACF).
- "pacf": Partial autocorrelation-based dissimilarity PACFDistance. Uses the **TSclust** package (see diss.PACF).
- "ar.lpc.ceps": Dissimilarity based on LPC cepstral coefficients ARLPCCepsDistance. Uses the **TSclust** package (see diss.AR.LPC.CEPS).
- "ar.mah": Model-based dissimilarity proposed by Maharaj (1996, 2000) ARMahDistance. Uses the **TSclust** package (see diss.AR.MAH).
- "ar.pic": Model-based dissimilarity measure proposed by Piccolo (1990) ARPicDistance. Uses the **TSclust** package (see diss.AR.PIC).
- "cdm": Compression-based dissimilarity measure CDMDistance. Uses the **TSclust** package (see diss.CDM).
- "cid": Complexity-invariant distance measure CIDDistance. Uses the **TSclust** package (see diss.CID).
- "cor": Dissimilarities based on Pearson's correlation CorDistance. Uses the **TSclust** package (see diss.COR).
- "cort": Dissimilarity index which combines temporal correlation and raw value behaviors CortDistance. Uses the **TSclust** package (see diss.CORT).
- "wav": Dissimilarity based on wavelet feature extraction WavDistance. Uses the **TSclust** package (see diss.DWT).
- "int.per": Integrated periodogram based dissimilarity IntPerDistance. Uses the **TSclust** package (see diss.INT.PER).
- "per": Periodogram based dissimilarity PerDistance. Uses the **TSclust** package (see diss.PER).
- "mindist.sax": Symbolic Aggregate Aproximation based dissimilarity MindistSaxDistance. Uses the **TSclust** package (see diss.MINDIST.SAX).
- "ncd": Normalized compression based distance NCDDistance. Uses the **TSclust** package (see diss.NCD).
- "pred": Dissimilarity measure cased on nonparametric forecasts PredDistance. Uses the **TSclust** package (see diss.PRED).

- "spec.glk": Dissimilarity based on the generalized likelihood ratio test SpecGLKDistance. Uses the **TSclust** package (see diss.SPEC.GLK).
- "spec.isd": Dissimilarity based on the integrated squared difference between the log-spectra SpecISDDistance. Uses the **TSclust** package (see diss.SPEC.ISD).
- "spec.llr": General spectral dissimilarity measure using local-linear estimation of the log-spectra SpecLLRDistance. Uses the **TSclust** package (see diss.SPEC.LLR).
- "pdc": Permutation Distribution Distance PDCDistance. Uses the **pdc** package (see pdcDist).
- "frechet": Frechet distance FrechetDistance. Uses the **longitudinalData** package (see distFrechet).

Some distance measures may require additional arguments. See the individual help pages (detailed above) for more information about each method. These parameters should be named in order to avoid mismatches.

Finally, for options `dissim`, `dissimapprox` and `sts`, databases conformed of series with different sampling rates can be introduced as a `list` of `zoo`, `xts` or `ts` objects, where each element in the list is a time series with its own time index.

Value

D	The computed distance matrix of the time series database. In some cases, such as ar.mahDistance or predDistance, some additional information is also provided.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

Examples

```
# The object example.database is a numeric matrix that saves
# 6 ARIMA time series in a row-wise format. For more information
# see help page of example.databases:

help(example.database)
data(example.database)

# To calculate the distance matrix of this database:

TSDatabaseDistances(example.database, distance="manhattan")
TSDatabaseDistances(example.database, distance="edr", epsilon=0.2)
TSDatabaseDistances(example.database, distance="fourier", n=20)

# The object zoo.database is a zoo object that saves
# the same 6 ARIMA time series saved in example.database.

data(zoo.database)

# To calculate the distance matrix of this database:

TSDatabaseDistances(zoo.database, distance="manhattan")
```

```
TSDatabaseDistances(zoo.database, distance="edr", epsilon=0.2)
TSDatabaseDistances(zoo.database, distance="fourier", n=20)
```

TSDistances *TSdist distance computation.*

Description

TSdist distance calculation between two time series.

Usage

```
TSDistances(x, y, tx, ty, distance, ...)
```

Arguments

x	Numeric vector or ts, zoo or xts object containing the first time series.
y	Numeric vector or ts, zoo or xts object containing the second time series.
tx	Optional temporal index of series x. Only necessary if x is a numeric vector and the sampling index is not constant.
ty	Optional temporal index of series y. Only necessary if y is a numeric vector and the sampling index is not constant.
distance	Distance measure to be used. It must be one of: "euclidean", "manhattan", "minkowski", "infnorm", "ccor", "sts", "dtw", "keogh.lb", "edr", "erp", "lcss", "fourier", "tquest", "dissim", "acf", "pacf", "ar.lpc.ceps", "ar.mah", "ar.mah.statistic", "ar.mah.pvalue", "ar.pic", "cdm", "cid", "cor", "cort", "wav", "int.per", "per", "mindist.sax", "ncd", "pred", "spec.glk", "spec.isd", "spec.llr", "pdc", "frechet")
...	Additional parameters required by the distance method.

Details

The distance between the two time series x and y is calculated. x and y can be saved in a numeric vector or a ts, zoo or xts object. The following distance methods are supported:

- "euclidean": Euclidean distance. EuclideanDistance
- "manhattan": Manhattan distance. ManhattanDistance
- "minkowski": Minkowski distance. MinkowskiDistance
- "infnorm": Infinite norm distance. InfNormDistance
- "ccor": Distance based on the cross-correlation. CCorDistance
- "sts": Short time series distance. STSDistance
- "dtw": Dynamic Time Warping distance. DTWDistance. Uses the **dtw** package (see dtw).
- "lb.keogh": LB_Keogh lower bound for the Dynamic Time Warping distance. LBKeoghDistance

- "edr": Edit distance for real sequences. EDRDistance
- "erp": Edit distance with real penalty. ERPDistance
- "lcss": Longest Common Subsequence Matching. LCSSDistance
- "fourier": Distance based on the Fourier Discrete Transform. FourierDistance
- "tquest": TQuest distance. TquestDistance
- "dissim": Dissim distance. DissimDistance
- "acf": Autocorrelation-based dissimilarity ACFDistance. Uses the **TSclust** package (see diss.ACF).
- "pacf": Partial autocorrelation-based dissimilarity PACFDistance. Uses the **TSclust** package (see diss.PACF).
- "ar.lpc.ceps": Dissimilarity based on LPC cepstral coefficients ARLPCCepsDistance. Uses the **TSclust** package (see diss.AR.LPC.CEPS).
- "ar.mah": Model-based dissimilarity proposed by Maharaj (1996, 2000) ARMahDistance. Uses the **TSclust** package (see diss.AR.MAH).
- "ar.pic": Model-based dissimilarity measure proposed by Piccolo (1990) ARPicDistance. Uses the **TSclust** package (see diss.AR.PIC).
- "cdm": Compression-based dissimilarity measure CDMDistance. Uses the **TSclust** package (see diss.CDM).
- "cid": Complexity-invariant distance measure CIDDistance. Uses the **TSclust** package (see diss.CID).
- "cor": Dissimilarities based on Pearson's correlation CorDistance. Uses the **TSclust** package (see diss.COR).
- "cort": Dissimilarity index which combines temporal correlation and raw value behaviors CortDistance. Uses the **TSclust** package (see diss.CORT).
- "wav": Dissimilarity based on wavelet feature extraction WavDistance. Uses the **TSclust** package (see diss.DWT).
- "int.per": Integrated periodogram based dissimilarity IntPerDistance. Uses the **TSclust** package (see diss.INT.PER).
- "per": Periodogram based dissimilarity PerDistance. Uses the **TSclust** package (see diss.PER).
- "mindist.sax": Symbolic Aggregate Aproximation based dissimilarity MindistSaxDistance. Uses the **TSclust** package (see diss.MINDIST.SAX).
- "ncd": Normalized compression based distance NCDDistance. Uses the **TSclust** package (see diss.NCD).
- "pred": Dissimilarity measure cased on nonparametric forecasts PredDistance. Uses the **TSclust** package (see diss.PRED).
- "spec.glk": Dissimilarity based on the generalized likelihood ratio test SpecGLKDistance. Uses the **TSclust** package (see diss.SPEC.GLK).
- "spec.isd": Dissimilarity based on the integrated squared difference between the log-spectra SpecISDDistance. Uses the **TSclust** package (see diss.SPEC.ISD).
- "spec.llr": General spectral dissimilarity measure using local-linear estimation of the log-spectra SpecLLRDistance. Uses the **TSclust** package (see diss.SPEC.LLR).

- "pdc": Permutation Distribution Distance PDCDistance. Uses the **pdc** package (see pdcDist).
- "frechet": Frechet distance FrechetDistance. Uses the **longitudinalData** package (see distFrechet).

Some distance measures may require additional arguments. See the individual help pages (detailed above) for more information about each method.

Value

d The computed distance between the pair of time series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

Examples

```
# The objects zoo.series1 and zoo.series2 are two
# zoo objects that save two series of length 100.

data(zoo.series1)
data(zoo.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# The distance calculation for these two series is done
# as follows:

TSDistances(zoo.series1, zoo.series2, distance="infnorm")
TSDistances(zoo.series1, zoo.series2, distance="cor", beta=3)
TSDistances(zoo.series1, zoo.series2, distance="dtw", sigma=20)
```

WavDistance *Dissimilarity for Time Series Based on Wavelet Feature Extraction*

Description

Provides the dissimilarity between two numerical series of the same length by calculating the Euclidean distance between the wavelet coefficients obtained from an orthogonal wavelet transform of the series.

Usage

```
WavDistance(x, y, ...)
```

Arguments

x	Numeric vector containing the first time series.
y	Numeric vector containing the second time series.
...	Additional parameters for the function. See diss.DWT for more information.

Details

This is essentially a wrapper for the diss.DWT function of package **TSclust**. However, in this case, instead of introducing a matrix conformed of a set of time series, two numerical series of the same length are introduced.

Value

d	The computed distance between the pair of series.

Author(s)

Usue Mori, Alexander Mendiburu, Jose A. Lozano.

References

Pablo Montero, José A. Vilar (2014). TSclust: An R Package for Time Series Clustering. Journal of Statistical Software, 62(1), 1-43. URL http://www.jstatsoft.org/v62/i01/.

See Also

To calculate this distance measure using `ts`, `zoo` or `xts` objects see TSDistances. To calculate distance matrices of time series databases using this measure see TSDatabaseDistances.

Examples

```
# The objects example.series1 and example.series2 are two
# numeric series of length 100.

data(example.series1)
data(example.series2)

# For information on their generation and shape see
# help page of example.series.

help(example.series)

# Calculate the distance based on wavelet feature extraction between the series.

WavDistance(example.series1, example.series2)
```

Index