Package 'rucrdtw'

October 13, 2017

Type Package

Title R Bindings for the UCR Suite

Version 0.1.3

Date 2017-10-12

BugReports https://github.com/pboesu/rucrdtw/issues

URL https://github.com/pboesu/rucrdtw

Description R bindings for functions from the UCR Suite by Rakthanmanon et al. (2012) <DOI:10.1145/2339530.2339576>, which enables ultrafast subsequence search for a best match under Dynamic Time Warping and Euclidean Distance.

License Apache License

LazyData TRUE

LinkingTo Rcpp

Depends R (>= 2.10)

Imports Rcpp

SystemRequirements C++11

RoxygenNote 6.0.1

Suggests testthat, knitr, rmarkdown, dtw, rbenchmark

VignetteBuilder knitr

NeedsCompilation yes

Author Philipp Boersch-Supan [aut, cre] (0000-0001-6723-6833),
Thanawin Rakthanmanon [aut],
Bilson Campana [aut],
Abdullah Mueen [aut],
Gustavo Batista [aut],
Eamonn Keogh [aut]

Maintainer Philipp Boersch-Supan <pboesu@gmail.com>

Repository CRAN

Date/Publication 2017-10-13 05:44:50 UTC

R topics documented:

`rucrdtw` *rucrdtw: Fast time series subsequence search in R*

Description

Dynamic Time Warping (DTW) methods provide algorithms to optimally map a given time series onto all or part of another time series. The remaining cumulative distance between the series after the alignement is a useful distance metric in time series data mining applications for tasks such as classification, clustering, and anomaly detection. A broad suite of DTW algorithms is implemented in R in the **dtw** package (Giorgino 2009).

Calculating a DTW alignment is computationally relatively expensive, and as a consequence DTW is often a bottleneck in time series data mining applications. The UCR Suite (Rakthanmanon et al. 2012) provides a highly optimized algorithm for best-match subsequence searches that avoids unnecessary distance computations and thereby enables fast DTW and Euclidean Distance queries even in data sets containing trillions of observations.

The **rucrdtw** package provides R bindings for the UCR Suite. In addition to queries and data stored in text files, rucrdtw also implements methods for queries and/or data that are held in memory as R objects, as well as a method to do fast similarity searches against reference libraries of time series. The following table gives a quick overview over the different methods provided by `rucrdtw`:

DTW method	ED method	data format	query format	Use case
`ucrdtw_ff`	`ucred_ff`	text file	text file	data sets that are too large to keep in memory
`ucrdtw_fv`	`ucred_fv`	text file	numeric vector	data sets that are too large to keep in memory
`ucrdtw_vv`	`ucred_vv`	numeric vector	numeric vector	data sets that fit into memory
`ucrdtw_mv`	`ucred_mv`	numeric matrix	numeric vector	compare query to a set of reference sequences of equal le

Examples of the functionality in this package are provided in the rucrdtw vignette:

```
vignette("using_rucrdtw", package = "rucrdtw")
```

References

Rakthanmanon, Thanawin, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 262-70. ACM. doi:10.1145/2339530.2339576.

Giorgino, Toni (2009). Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. Journal of Statistical Software, 31(7), 1-24, doi:10.18637/jss.v031.i07.

UCR Suite Website: http://www.cs.ucr.edu/~eamonn/UCRsuite.html

See Also

dtw-package

`summary.ucrdtw` *Summarize subsequence search*

Description

Summary method for class ucrdtw

Usage

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

Arguments

`object`	an object of class ucrdtw
`...`	further arguments passed to or from methods

Value

An unlisted version of the object is returned.

summary.ucred *Summarize subsequence search*

Description

Summary method for class ucred

Usage

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

Arguments

object	an object of class ucred
...	further arguments passed to or from methods

Value

An unlisted version of the object is returned.

synthetic_control *Synthetic Control Chart Time Series Data Set*

Description

This dataset contains 600 examples of control charts synthetically generated by the process in Alcock and Manolopoulos (1999) and obtained from the UCI Machine Learning Repository http://archive.ics.uci.edu/ml/datasets/Synthetic+Control+Chart+Time+Series.

Format

A matrix with 600 rows (series) and 60 columns (timepoints)

Details

There are six different classes of control charts:

1. Normal (Rows 1-100)
2. Cyclic (Rows 101-200)
3. Increasing trend (Rows 201-300)
4. Decreasing trend (Rows 301-400)
5. Upward shift (Rows 401-500)
6. Downward shift (Rows 501-600)

References

Alcock R.J. and Manolopoulos Y. Time-Series Similarity Queries Employing a Feature-Based Approach. 7th Hellenic Conference on Informatics. August 27-29. Ioannina, Greece 1999.

`ucrdtw_ff` *UCR DTW Algorithm file-file method*

Description

Sliding-window similarity search using DTW distance. This implementation is very close to the UCR Suite command line utility, in that it takes files as inputs for both query and data

Usage

```
ucrdtw_ff(data, query, qlength, dtwwindow)
```

Arguments

`data`	character; path to data file
`query`	character; path to query file
`qlength`	integer; length of the query (number of data points), usually this should be equivalent to length(query), but it can be shorter.
`dtwwindow`	double; Size of the warping window size (as a proportion of query length). The DTW calculation in 'rucrdtw' uses a symmetric Sakoe-Chiba band. See Giorgino (2009) for a general coverage of warping window constraints.

Value

a ucrdtw object. A list with the following elements

- **location:** The starting location of the nearest neighbor of the given query, of size `length(query)`, in the data. Note that location starts from 1.
- **distance:** The DTW distance between the nearest neighbor and the query.
- **prunedKim:** Percentage of subsequences that were pruned based on the LB-Kim criterion.
- **prunedKeogh:** Percentage of subsequences that were pruned based on the LB-Keogh-EQ criterion.
- **prunedKeogh2:** Percentage of subsequences that were pruned based on the LB-Keogh-EC criterion.
- **dtwCalc:** Percentage of subsequences for which the full DTW distance was calculated.

For an explanation of the pruning criteria see Rakthanmanon et al. (2012).

References

Rakthanmanon, Thanawin, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 262-70. ACM. doi:10.1145/2339530.2339576.

Giorgino, Toni (2009). Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. Journal of Statistical Software, 31(7), 1-24, doi:10.18637/jss.v031.i07.

Examples

```
#locate example data file
dataf <- system.file("extdata/col_sc.txt", package="rucrdtw")
#locate example query file
queryf <- system.file("extdata/mid_sc.txt", package="rucrdtw")
#determine length of query file
qlength <- length(scan(queryf))
#run query
ucrdtw_ff(dataf, queryf, qlength, 0.05)
```

`ucrdtw_fv` *UCR DTW Algorithm file-vector method*

Description

Sliding-window similarity search using DTW distance. This implementation of the UCR Suite command line utility, takes a data file as input and an R numeric vector for the query

Usage

```
ucrdtw_fv(data, query, dtwwindow)
```

Arguments

`data`	character; path to data file
`query`	numeric vector containing the query. The query length is set to the length of this object.
`dtwwindow`	double; Size of the warping window size (as a proportion of query length). The DTW calculation in ‘rucrdtw‘ uses a symmetric Sakoe-Chiba band. See Giorgino (2009) for a general coverage of warping window constraints.

Value

a ucrdtw object. A list with the following elements

- **location:** The starting location of the nearest neighbor of the given query, of size `length(query)`, in the data. Note that location starts from 1.
- **distance:** The DTW distance between the nearest neighbor and the query.
- **prunedKim:** Percentage of subsequences that were pruned based on the LB-Kim criterion.
- **prunedKeogh:** Percentage of subsequences that were pruned based on the LB-Keogh-EQ criterion.
- **prunedKeogh2:** Percentage of subsequences that were pruned based on the LB-Keogh-EC criterion.
- **dtwCalc:** Percentage of subsequences for which the full DTW distance was calculated.

For an explanation of the pruning criteria see Rakthanmanon et al. (2012).

References

Rakthanmanon, Thanawin, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 262-70. ACM. doi:10.1145/2339530.2339576.

Giorgino, Toni (2009). Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. Journal of Statistical Software, 31(7), 1-24, doi:10.18637/jss.v031.i07.

Examples

```
#locate example data file
dataf <- system.file("extdata/col_sc.txt", package="rucrdtw")
#load example data set
data("synthetic_control")
#extract first row as query
query <- synthetic_control[1,]
#run query
ucrdtw_fv(dataf, query, 0.05)
```

`ucrdtw_mv` *UCR DTW Algorithm matrix-vector method*

Description

This implementation of the UCR Suite command line utility, takes an R numeric matrix as data input and an R numeric vector for the query. The default behaviour differs from the other methods, in that it does not perform a sliding window search for a match. Instead it is designed to find a best match for a query in a reference set of time-series of the same length as the query. This is useful, for example, when comparing a time-series of undetermined class to a labelled reference set of classified time-series.

Usage

```
ucrdtw_mv(data, query, dtwwindow, epoch = 1e+05, skip = TRUE,
  byrow = FALSE)
```

Arguments

data	numeric matrix containing data
query	numeric vector containing the query. This determines the query length.
dtwwindow	double; Size of the warping window size (as a proportion of query length). The DTW calculation in 'rucrdtw' uses a symmetric Sakoe-Chiba band. See Giorgino (2009) for a general coverage of warping window constraints.
epoch	int defaults to 1e5, should be $\leq$ 1e6. This is the size of the data chunk that is processed at once. All cumulative values in the algorithm will be restarted after epoch iterations to reduce floating point errors in these values.
skip	boolean; defaults to TRUE. If TRUE bound calculations and if necessary, distance calculations, are only performed on non-overlapping segments of the data (i.e. multiples of `length(query)`). This is useful if `data` is a set of multiple reference time series, each of length `length(query)`. The location returned when skipping is the index of the subsequence.
byrow	logical; If TRUE rows in `data` represent time-series, columns time-points

Value

a ucrdtw object. A list containing the following elements

- **location:** The row or column number of the nearest neighbor of the given query in the data set.
- **distance:** The DTW distance between the nearest neighbor and the query.
- **prunedKim:** Percentage of subsequences that were pruned based on the LB-Kim criterion.
- **prunedKeogh:** Percentage of subsequences that were pruned based on the LB-Keogh-EQ criterion.
- **prunedKeogh2:** Percentage of subsequences that were pruned based on the LB-Keogh-EC criterion.
- **dtwCalc:** Percentage of subsequences for which the full DTW distance was calculated.

References

Giorgino, Toni (2009). Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. Journal of Statistical Software, 31(7), 1-24, doi:10.18637/jss.v031.i07.

Examples

```
#load example data
data("synthetic_control")
query <- synthetic_control[5,]
#run query
ucrdtw_mv(synthetic_control, query, 0.05, byrow = TRUE)
```

ucrdtw_vv *UCR DTW Algorithm vector-vector method*

Description

Sliding-window similarity search using DTW distance. This implementation of the UCR Suite command line utility, takes an R numeric vector as data input and an R numeric vector for the query

Usage

```
ucrdtw_vv(data, query, dtwwindow, epoch = 100000L, skip = FALSE)
```

Arguments

`data`	numeric vector containing data
`query`	numeric vector containing the query. The length of this vector determines the query length.
`dtwwindow`	double; Size of the warping window size (as a proportion of query length). The DTW calculation in 'rucrdtw' uses a symmetric Sakoe-Chiba band. See Giorgino (2009) for a general coverage of warping window constraints.
`epoch`	int defaults to 1e5, should be $\leq$ 1e6. This is the size of the data chunk that is processed at once. All cumulative values in the algorithm will be restarted after `epoch` iterations to reduce floating point errors in these values.
`skip`	bool defaults to FALSE. If TRUE bound calculations and if necessary, distance calculations, are only performed on non-overlapping segments of the data (i.e. multiples of `qlength`). This is useful if `data` is a set of multiple reference time series, each of length `qlength`. The location returned when skipping is the index of the subsequence.

Value

a ucrdtw object. A list with the following elements

- **location:** The starting location of the nearest neighbor of the given query, of size `length(query)`, in the data. Note that location starts from 1.
- **distance:** The DTW distance between the nearest neighbor and the query.
- **prunedKim:** Percentage of subsequences that were pruned based on the LB-Kim criterion.
- **prunedKeogh:** Percentage of subsequences that were pruned based on the LB-Keogh-EQ criterion.
- **prunedKeogh2:** Percentage of subsequences that were pruned based on the LB-Keogh-EC criterion.
- **dtwCalc:** Percentage of subsequences for which the full DTW distance was calculated.

For an explanation of the pruning criteria see Rakthanmanon et al. (2012).

References

Rakthanmanon, Thanawin, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 262-70. ACM. doi:10.1145/2339530.2339576.

Giorgino, Toni (2009). Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. Journal of Statistical Software, 31(7), 1-24, doi:10.18637/jss.v031.i07.

Examples

```
#read example data into vector
datav <- scan(system.file("extdata/col_sc.txt", package="rucrdtw"))
#read example query into vector
query <- scan(system.file("extdata/first_sc.txt", package="rucrdtw"))
#execute query
ucrdtw_vv(datav, query, 0.05)
```

`ucred_ff` *UCR ED Algorithm file-file method*

Description

Sliding-window similarity search using euclidean distance. This implementation is very close to the UCR Suite command line utility, in that it takes files as inputs for both query and data

Usage

```
ucred_ff(data, query, qlength)
```

Arguments

`data`	character; path to data file
`query`	character; path to query file
`qlength`	integer; length of query (n data points). Usually the length of the data contained in `query`, but it can be shorter.

Value

a ucred object. A list with the following elements

- **location:** The starting location of the nearest neighbor of the given query, of size `qlength`, in the data. Note that location starts from 1.
- **distance:** The euclidean distance between the nearest neighbor and the query.

Examples

```
#locate example data file
dataf <- system.file("extdata/col_sc.txt", package="rucrdtw")
#locate example query file
queryf <- system.file("extdata/mid_sc.txt", package="rucrdtw")
#determine length of query file
qlength <- length(scan(queryf))
#run query
ucred_ff(dataf, queryf, qlength)
```

---

ucred_fv *UCR ED Algorithm file-vector method*

Description

Sliding-window similarity search using Euclidean Distance. This implementation of the UCR Suite command line utility, takes a data file as input and an R numeric vector for the query.

Usage

```
ucred_fv(data, query)
```

Arguments

data	character; path to data file
query	numeric vector containing query

Value

a ucred object. A list with the following elements

- **location:** The starting location of the nearest neighbor of the given query, of size `length(query)`, in the data. Note that location starts from 1.
- **distance:** The Euclidean Distance between the nearest neighbor and the query.

Examples

```
#locate example data file
dataf <- system.file("extdata/col_sc.txt", package="rucrdtw")
#read example query file into vector
query <- scan(system.file("extdata/mid_sc.txt", package="rucrdtw"))
#run query
ucred_fv(dataf, query)
```

ucred_mv *UCR ED Algorithm matrix-vector method*

Description

This implementation of the UCR Suite command line utility, takes an R numeric matrix as data input and an R numeric vector for the query. The default behaviour differs from the other methods, in that it does not perform a sliding window search for a match. Instead it is designed to find a best match for a query in a reference set of time-series of the same length as the query. This is useful, for example, when comparing a time-series of undetermined class to a labelled reference set of classified time-series.

Usage

```
ucred_mv(data, query, skip = TRUE, byrow = FALSE)
```

Arguments

data	numeric matrix containing data
query	numeric vector containing the query. This determines the query length.
skip	boolean; defaults to TRUE. If TRUE bound calculations and if necessary, distance calculations, are only performed on non-overlapping segments of the data (i.e. multiples of `length(query)`). This is useful if `data` is a set of multiple reference time series, each of length `length(query)`. The location returned when skipping is the index of the subsequence.
byrow	logical; If TRUE rows in `data` represent time-series, columns time-points

Value

a ucred object. A list with the following elements

- **location:** The starting location of the nearest neighbor of the given query, of size `qlength`, in the data. Note that location starts from 1.
- **distance:** The euclidean distance between the nearest neighbor and the query.

Examples

```
#load example data matrix
data("synthetic_control")
#use on arbitrary row as query
query <- synthetic_control[5,]
#run query
ucred_mv(synthetic_control, query, byrow=TRUE)
```

ucred_vv *UCR ED Algorithm vector-vector method*

Description

Sliding-window similarity search using Euclidean Distance. This implementation of the UCR Suite Euclidean Distance command line utility takes an R numeric vector as data input and an R numeric vector for the query.

Usage

```
ucred_vv(data, query, skip = FALSE)
```

Arguments

`data`	numeric vector containing data
`query`	numeric vector containing query
`skip`	bool defaults to TRUE. If TRUE bound calculations and if necessary, distance calculations, are only performed on non-overlapping segments of the data (i.e. multiples of `length(query)`). This is useful if `data` is a set of multiple reference time series, each of length `length(query)`. The location returned when skipping is the index of the subsequence.

Value

a ucred object. A list with the following elements

- **location:** The starting location of the nearest neighbor of the given query, of size `length(query)`, in the data. Note that location starts from 1.
- **distance:** The Euclidean Distance between the nearest neighbor and the query.

Examples

```
#read example file into vector
dataf <- scan(system.file("extdata/col_sc.txt", package="rucrdtw"))
#read example query file into vector
query <- scan(system.file("extdata/mid_sc.txt", package="rucrdtw"))
#run query
ucred_vv(dataf, query)
```

Index