Package 'TSMining'

June 26, 2015

Type Package

Title Mining Univariate and Multivariate Motifs in Time-Series Data

Description Implementations of a number of functions used to mine numeric time-series data. It covers the implementation of SAX transformation, univariate motif discovery (based on the random projection method), multivariate motif discovery (based on graph clustering), and several functions used for the ease of visualizing the motifs discovered. The details of SAX transformation can be found in J. Lin. E. Keogh, L. Wei, S. Lonardi, Experiencing SAX: A novel symbolic representation of time series, Data Mining and Knowledge Discovery 15 (2) (2007) 107-144. Details on univariate motif discovery method implemented can be found in B. Chiu, E. Keogh, S. Lonardi, Probabilistic discovery of time series motifs, ACM SIGKDD, Washington, DC, USA, 2003, pp. 493-498. Details on the multivariate motif discovery method implemented can be found in A. Vahdatpour, N. Amini, M. Sarrafzadeh, Towards unsupervised activity discovery using multi-dimensional motif detection in time series, IJCAI 2009 21st International Joint Conference on Artificial Intelligence.

Version 1.0

Date 2015-06-24

Author Cheng Fan

Maintainer Cheng Fan <raja8885@hotmail.com>

Depends R (>= 3.0.1)

Imports foreach, ggplot2, plyr, reshape2

Suggests knitr

VignetteBuilder knitr

LazyData true

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2015-06-26 00:02:41

R topics documented:

`TSMining-package` *Implementation of Univariate and Multivariate Motif Discovery in Time-Series Data*

Description

Implementations of a number of functions used to mine numeric time-series data. It covers the implementation of SAX transformation, univariate motif discovery (based on the random projection method), multivariate motif discovery (based on graph clustering), and several functions used for the ease of visualizing the motifs discovered. The details of SAX transformation can be found in J. Lin. E. Keogh, L. Wei, S. Lonardi, Experiencing SAX: A novel symbolic representation of time series, Data Mining and Knowledge Discovery 15 (2) (2007) 107-144. Details on univariate motif discovery method implemented can be found in B. Chiu, E. Keogh, S. Lonardi, Probabilistic discovery of time series motifs, ACM SIGKDD, Washington, DC, USA, 2003, pp. 493-498. Details on the multivariate motif discovery method implemented can be found in A. Vahdatpour, N. Amini, M. Sarrafzadeh, Towards unsupervised activity discovery using multi-dimensional motif detection in time series, IJCAI 2009 21st International Joint Conference on Artificial Intelligence.

Details

Package:	TSMining
Type:	Package
Version:	1.0
Date:	2015-06-24
License:	GPL-3

Author(s)

Cheng Fan

Maintainer: Cheng Fan <raja8885@hotmail.com> Cheng Fan

References

J. Lin. E. Keogh, L. Wei, S. Lonardi. Experiencing SAX: A novel symbolic representation of time series. Data Mining and Knowledge Discovery 15 (2) (2007) 107-144. DOI:http://cs.gmu.edu/~jessica/SAX_DAMI_preprint

B. Chiu, E. Keogh, S. Lonardi. Probabilistic discovery of time series motifs. ACM SIGKDD, Washington, DC, USA, 2003, pp. 493-498. DOI:10.1145/956750.956808

A. Vahdatpour, N. Amini, M. Sarrafzadeh. Towards unsupervised activity discovery using multi-dimensional motif detection in time series. IJCAI 2009 21st International Joint Conference on Artificial Intelligence. DOI:http://ijcai.org/papers09/Papers/IJCAI09-212.pdf

J. Buhler, M. Tompa. Finding motifs using random projections. 5th Annual International Conference on Computational Molecular Biology, ACM New York, New York, USA, 2001, pp. 69-76. DOI:http://www.cs.columbia.edu/~cleslie/cs4761/papers/buhler_finding.pdf

---

`BuildOperation` *An example 1-week data set containing the power consumption data of two building services sub-systems*

Description

The data contain 7-day power consumption data of two building services sub-systems, i.e., water-cooled chillers (WCC) and air handling unit (AHU) The collection interval is 15-minute. The data have 672 observations and 6 variables, i.e., Month, Day, Hour, Minute, WCC and AHU

Usage

```
data(BuildOperation)
```

Format

A data frame with 672 rows and 6 variables

Examples

```
library(ggplot2)
data(BuildOperation)
ggplot(data = BuildOperation, aes(x = 1:dim(BuildOperation)[1], y = WCC)) +
 geom_line() + geom_point()
ggplot(data = BuildOperation, aes(x = 1:dim(BuildOperation)[1], y = AHU)) +
 geom_line() + geom_point()
```

Func.dist *A function to calculate the distance between two SAX representations*

Description

This function calculates the distance between two SAX representations

Usage

```
Func.dist(x, y, mat, n)
```

Arguments

x	is a SAX representations.
y	is a SAX representations. It should have the same length as x.
mat	is the distance matrix created by Func.matrix
n	is the length of the original time series before the SAX transformation

Value

The function returns a numeric value, which is the distance between two SAX representations

Examples

```
#Assuming the original time series has a length of 20, n=20
#Assuming the time series is transformed into SAX representations using w=4 and a=4
#Assuming one is a,b,c,d and the other is d,b,c,d
Func.dist(x=c("a","b","c","d"), y=c("d","b","c","d"), mat=Func.matrix(a=4), n=20)
```

Func.matrix *A function to create the distance matrix for alphabets*

Description

This function create a distance matrix for alphabets used for SAX transformation

Usage

```
Func.matrix(a)
```

Arguments

a	is an integer specifying the alphabet size.

Value

The function returns a matrix showing the distance between alphabets

Examples

```
Func.matrix(a=5)
```

`Func.motif` *A function implementing the univaraite motif discovery algorithm using random projection*

Description

The function implements the univariate motif discovery algorithm proposed in B. Chiu, E. Keogh, S. Lonardi. Probabilistic discovery of time series motifs. ACM SIGKDD, Washington, DC, USA, 2003, pp. 493-498.

Usage

```
Func.motif(ts, global.norm, local.norm, window.size, overlap, w, a, mask.size,
  eps = 0.1, iter = 25, max.dist.ratio = 1.2, count.ratio.1 = 1.5,
  count.ratio.2 = 1.2)
```

Arguments

`ts`	is a numeric vector representing the univarate time series
`global.norm`	is a logical value specifying whether global standardization should be used for the whole time series
`local.norm`	is a logical value specifying whether local standardization should be used for each subsequences
`window.size`	is a integer which defines the length of the sliding window used to create subsequences
`overlap`	is a numeric value ranging from 0 to 1. It defines the percentage of overlapping when using sliding window to create subsquences. 0 means subsequences are created without overlaps. 1 means subsequences are created with the maximum overlap possible.
`w`	is an integer which defines the word size used for SAX transformation
`a`	is an integer which defines the alphabet size used for SAX transformation
`mask.size`	is the mask size used for random projection. It should be an integer ranging from 1 to the word size w
`eps`	is the minimum threshold for variance in subsequence and should be a numeric value. If the subsequence considered has a smaller variance than eps, it will be represented as a word using the middle alphabet. The default value is 0.1

iter	is an integer which specifies the iteration number in random projection, default value is 25
max.dist.ratio	is a numeric value used to add other possible members to a motif candidate. Default value is 1.2. Each motif candidate has two subsequences. The distance between these two candidates are calculated as a baseline, denoted as BASE. Any subsequence, whose distance to the motif candidate is smaller than max.dist.ratio*BASE, is considered as a member of that motif candidate.
count.ratio.1	defines the ratio between the iteration number and the minimum value in the collision matrix to be considered as motif candidate. Default value is 1.5. For instance, if the iter is 100, any pair of subsequence, which results in a value larger than 67 in the collision matrix, is considered as a motif candidate.
count.ratio.2	defines the ratio between the maximum counts in the collision matrix and any other count values that will be considered as potential members to a motif candidate

Value

The function returns a list of 6 elements. The first element is Subs, which is a data frame containing all the subsequences in original data formatThe second element is Subs.SAX, which is a data frame containing all the subsequences in SAX representations. The third element is Motif.raw, which is a list showing the motifs discovered in original data format.The fourth element is Motif.SAX, which is a list showing the motifs discovered in SAX representations. The fifth element is Collision.matrix, which is matrix containing the results of random projection. The sixth element is Indices, which is a list showing the starting positions of subsequences for each motif discovered.

Examples

```
#Perform the motif discovery for the first time series in the example data
data(test)
res.1 <- Func.motif(ts = test$TS1, global.norm = TRUE, local.norm = FALSE,
window.size = 10, overlap = 0, w = 5, a = 3, mask.size = 3, eps = .01)
#Check the number of motifs discovered
length(res.1$Indices)
#Check the starting positions of subsequences of each motif discovered
res.1$Indices
```

Func.motif.multivariate

A function to implment the multivariate motif discovery

Description

This function implements the multivariate motif discovery method proposed in A. Vahdatpour, N. Amini, M. Sarrafzadeh. Towards unsupervised activity discovery using multi-dimensional motif detection in time series. IJCAI 2009 21st International Joint Conference on Artificial Intelligence.

Usage

```
Func.motif.multivariate(motif.list, window.sizes, alpha)
```

Arguments

motif.list	is a list of lists, each contains the univariate motifs discovered in a univariate time series. The component of motif.list is the results of Func.motif()$Indices, which store the starting position of subsequences of each univariate motif
window.sizes	is a vector containing the length of motifs in each univariate time series. It should have the same order as components in motif.list.
alpha	is a numeric ranging from 0 to 1. It specifies the minimum correlation between two univariate motifs before considered as a multivariate motifs

Value

The function returns a list containing two elements. The first element is Motif, which is a list containing the univarate motif IDs for different multivariate motifs. e.g., if there are two univariate time series and each has 3 motifs, then univariate ID is from 1 to 6. The second element is Info, which is a list storing the information of univariate motifs for different multivariate motifs

Examples

```
data(test)
#Perform univariate motif discovery for each dimension in the example data
res.1 <- Func.motif(ts = test$TS1, global.norm = TRUE, local.norm = FALSE,
window.size = 10, overlap = 0, w = 5, a = 3, mask.size = 3, eps = .01)
res.2 <- Func.motif(ts = test$TS2, global.norm = TRUE, local.norm = FALSE,
window.size = 20, overlap = 0, w = 5, a = 3, mask.size = 3, eps = .01)
#Perform multivaraite motif discovery
res.multi <- Func.motif.multivariate(motif.list = list(res.1$Indices, res.2$Indices),
window.sizes = c(10,20), alpha = .8)
```

Func.SAX *A function to perform symbolic approximation aggregate (SAX) for time series data*

Description

The function create SAX symbols for a univariate time series. The details of this method can be referred to J. Lin, E. Keogh, L. Wei, S. Lonardi. Experiencing SAX: a novel symbolic representation of time series

Usage

```
Func.SAX(x, w, a, eps, norm)
```

Arguments

x	is a numeric vector representing the univariate time series
w	is the word size and should be an integer
a	is the alphabet size and should be an integer
eps	is the minimum threshold for variance in x and should be a numeric value. If x has a smaller variance than eps, it will represented as a word using the middle alphabet.
norm	is a logical value deciding whether standardization should be applied to x. If True, x is standardized using mean and standard deviation

Value

The function returns a SAX representation of x

Examples

```
x <- runif(n = 20, min = 0, max = 20)
Func.SAX(x = x, w = 5, a = 5, eps = .01, norm = TRUE)
```

Func.visual.MultiMotif

A function to prepare the data for the visualization of multivariate motifs discovered

---

Description

This function prepares the data used for visualizing multivariate motifs.

Usage

```
Func.visual.MultiMotif(data, multi.motifs, index)
```

Arguments

data	is a data frame containing the multivariate time series data. Each column represents a time series.
multi.motifs	is the result of Func.motif.multivariate
index	is an integer which specifies the No. of multivariate motif to be plotted

Value

The function returns a data frame for the ease of visualizing multivariate motif discovered

Examples

```
data(test)
#Perform univariate motif discovery
res.1 <- Func.motif(ts = test$TS1, global.norm = TRUE, local.norm = FALSE,
window.size = 10, overlap = 0, w = 5, a = 3, mask.size = 3, eps = .01)
res.2 <- Func.motif(ts = test$TS2, global.norm = TRUE, local.norm = FALSE,
window.size = 20, overlap = 0, w = 5, a = 3, mask.size = 3, eps = .01)
res.multi <- Func.motif.multivariate(motif.list = list(res.1$Indices, res.2$Indices),
window.sizes = c(10,20), alpha = .8)
#Use the function to prepare the data frame for visualizing the first multivariate motifs identified
data.multi <- Func.visual.MultiMotif(data = test, multi.motifs = res.multi, index = 1)
#Make the plot using ggplot2
library(ggplot2)
ggplot(data = data.multi) +
 geom_line(aes(x = T, y = X)) +
 geom_point(aes(x = T, y = X, col=Lab, shape=Lab)) + facet_grid(Facet~.)
```

Func.visual.SingleMotif

A function to prepare the dataset for visualizing the univaraite motifs discovered

---

Description

This function create a data set for the use of visualizing the univariate motifs discovered

Usage

```
Func.visual.SingleMotif(single.ts, window.size, motif.indices)
```

Arguments

single.ts	is a numeric vector used to represent the univariate time series
window.size	is the window size used to create subsequences. It is also the length of univariate motifs
motif.indices	is the results of Func.motif()$Indices, which store the starting position of subsequences for each univaraite motifs

Value

The function returns a list of three elements. The first element is data.1, which can be used to show the whole time series with motifs identifed highlighted. The second element is data.2, which can be used to visualize the members of each motif. It is a list containing data frames. Each data frame is designed to visualize the members in each motif.

Examples

```
data(test)
#Perform univariate motif discovery for the first dimension data in the example data
res.1 <- Func.motif(ts = test$TS1, global.norm = TRUE, local.norm = FALSE,
window.size = 10, overlap = 0, w = 5, a = 3, mask.size = 3, eps = .01)
data.vis <- Func.visual.SingleMotif(single.ts=test$TS1, window.size=10, motif.indices=res.1$Indices)
#To visualize general information of motifs discovered on the whole time series
library(ggplot2)
ggplot(data = data.vis$data.1) +
 geom_line(aes(x = 1:dim(data.vis$data.1)[1], y = X)) +
 geom_point(aes(x = 1:dim(data.vis$data.1)[1], y = X, color=Y))
#To visualize the detailed information of the 1st motif
ggplot(data = data.vis$data.2[[1]]) + geom_line(aes(x = Time, y = Value, linetype=Instance))
```

---

test *An example data set for univariate motif discovery*

Description

The data is a data frame containing 100 observations and 2 variables. The first time series is denoted as TS1. It is created in such a way that two motifs are embedded, each with two appearances and a length of 10. The rest are randomly generated. The second time series is denoted as TS2. It is created in such a way that one motif with three appearances are embedded. It has a length of 20. The rest are randomly generated. This synthetic data set is used as examples for motif discovery

Usage

```
data(test)
```

Format

A data frame with 100 rows and 2 variables

Examples

```
library(ggplot2)
data(test)
ggplot(data = test, aes(x = 1:dim(test)[1], y = TS1)) + geom_line() + geom_point()
ggplot(data = test, aes(x = 1:dim(test)[1], y = TS2)) + geom_line() + geom_point()
```

Index