Package ‘QWDAP’

April 19, 2024

Type Package

Title Quantum Walk-Based Data Analysis and Prediction

Version 1.1.18

Date 2024-04-20

Author Binghuang Pan [aut, cre],
Zhaoyuan Yu [aut],
Xu Hu [ctb],
Yuhao Teng [ctb]

Maintainer Binghuang Pan <bright1up@163.com>

Description The modeling and prediction of graph-associated time series(GATS) based on continuous time quantum walk. This software is mainly used for feature extraction, modeling, prediction and result evaluation of GATS, including continuous time quantum walk simulation, feature selection, regression analysis, time series prediction, and series fit calculation.

Imports pls, CORElearn, Rcpp, methods

LinkingTo Rcpp,RcppEigen

License GPL-2

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.1.2

Depends R (>= 3.5.0)

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-04-19 12:12:43 UTC

R topics documented:

qwdap.eval 2
qwdap.pcr 3
qwdap.plsr 3

qwdap.ppr 4
qwdap.predict 5
qwdap.qwalk 5
qwdap.rrelieff 6
qwdap.swr 7
qwdap.sws 8
qwdap.var 9
traffic.model.n1 9
traffic.n1 10
traffic.qw 10
trafficflow 11

Index **12**

`qwdap.eval` *Evaluation*

Description

calculate the Coefficient of Determination, Root Mean Squared Error and the Mean Absolute Error between two series.

Usage

```
qwdap.eval(series1, series2)
```

Arguments

`series1`	The series1.
`series2`	The series2.

Value

Three indicators, the Coefficient of Determination, Root Mean Squared Error and Mean Absolute Error.

Examples

```
set.seed(1)
res.eval <- qwdap.eval(rnorm(100,0,2),rnorm(100,0,1))
```

qwdap.pcr *Principle Component Regression*

Description

Principle component regression. This is a linear regression method used to establish the linear relationship between the original time series and the modes generated by quantum walks.

Usage

```
qwdap.pcr(in_data, data_range, plotting)
```

Arguments

`in_data`	a 'QWMS' object, which includes the target series and the selected modes which can be obtained from modes selection.
`data_range`	the range of the train samples.
`plotting`	whether to plot.

Value

a 'QWMODEL' object which includes the information of regression analysis.

Examples

```
data("traffic.n1")
res.pcr <- qwdap.pcr(traffic.n1,c(1,500), FALSE)
```

qwdap.plsr *Partial Least Squares Regression*

Description

Partial least squares regression. This is a linear regression method used to establish the linear relationship between the original time series and the modes generated by quantum walks.

Usage

```
qwdap.plsr(in_data, data_range, plotting)
```

Arguments

`in_data`	a 'QWMS' object, which includes the target series and the selected modes which can be obtained from modes selection.
`data_range`	the range of the train samples.
`plotting`	whether to plot.

Value

a 'QWMODEL' object which includes the information of regression analysis.

Examples

```
data("traffic.n1")
res.plsr <- qwdap.plsr(traffic.n1,c(1,500),FALSE)
```

---

qwdap.ppr *Projection Pursuit Regression*

Description

Projection pursuit regression. This is a nonlinear regression method used to establish the nonlinear relationship between the original time series and the modes generated by quantum walks.

Usage

```
qwdap.ppr(in_data, data_range, plotting)
```

Arguments

`in_data`	a 'QWMS' object, which includes the target series and the selected modes which can be obtained from modes selection.
`data_range`	the range of the train samples.
`plotting`	whether to plot.

Value

a 'QWMODEL' object which includes the information of regression analysis.

Examples

```
data("traffic.n1")
res.ppr <- qwdap.ppr(traffic.n1,c(1,500))
```

qwdap.predict *Prediction*

Description

Based on the established model, make predict. The core algorithm of VAR prediction comes from MTS(ver. 1.1.1).

Usage

```
qwdap.predict(in_model, data_range)
```

Arguments

in_model	a 'QWMODEL' object, which is the model built by Stepwise Regression, PCR, PLSR, PPR, VAR in this package.
data_range	indicate the index range of the part data generated by quantum walks for predict.

Value

the predict data.

Examples

```
data(traffic.model.n1)
res.predict <- qwdap.predict(traffic.model.n1,c(501,720))
```

qwdap.qwalk *Quantum Walk*

Description

Generate the modes, the probabilities that the walker being found at vertices. An adjacency matrix is need for the process.

Usage

```
qwdap.qwalk(edges, startindex, lens, scals, getfloat)
```

Arguments

edges	your N*N adjacency matrix saved as list.
startindex	the initial position of the quantum walker.
lens	the number of records required in a round of sampling by a scaling factor. Set the length of the series according to requirements.
scals	the scaling factors used.
getfloat	Whether to return floating point data.

Details

'qwdap.qwalk()' is used to generated modes for time series analysis, the result is a object of class 'CTQW', the modes are saved in the object as a 3-dim array, and the parameters are also store in the object. The continuous time quantum walk is a continuous process, the modes are generated with a series of times, the parameter 'scals' can be understood as the tolerance of the arithmetic time series. Multiply tolerances can be passed in to obtain modes on different time scales through parameter 'scals'. The probability of the series with the probabilities that the walker being found at the vertices, and the length depends on parameter 'lens'. The data generated by this function is not recorded from the initial state. The shortest distance between all vertices and the initial position of the quantum walker is obtained by the Dijkstra algorithm.The probabilities corresponding to each vertex are recorded starting from the vertex furthest in the shortest distance is not 0. The function is single thread.

Value

a object of class 'CTQW', the quantum walk results and some parameters.

Author(s)

Pan Binghuang

Examples

```
edges <- matrix(c(0,1,0,0,0,0,0,
                  1,0,1,0,0,0,0,
                  0,1,0,1,0,0,0,
                  0,0,1,0,1,0,0,
                  0,0,0,1,0,1,0,
                  0,0,0,0,1,0,1,
                  0,0,0,0,0,1,0),
                nrow = 7)
res.qwalk <- qwdap.qwalk(edges,1,100,scals=seq(from=0.01, by=0.01, length.out=5))
```

`qwdap.rrelieff` *RReliefF*

Description

Mode selection by RReliefF. The purpose of this function is to select the part modes with similar characteristics to the observed time series from the modes generated by the quantum walk. And it is based on the data model.

Usage

```
qwdap.rrelieff(real, ctqw, index, num, plotting)
```

Arguments

`real`	the real series observed.
`ctqw`	the 'CTQW' object.
`index`	the index of the data for mode selection.
`num`	the number of series required.
`plotting`	whether to plot.

Details

The 'QWMS' object include the original time series and the modes generated by quantum walks.

Value

a 'QWMS' object.

Examples

```
data("traffic.qw")
data("trafficflow")
res.rrelieff <- qwdap.rrelieff(trafficflow,traffic.qw,1,30,TRUE)
```

---

qwdap.swr *Model by Stepwise Regression*

Description

Stepwise regression. This is a linear regression method used to establish the linear relationship between the original time series and the modes generated by quantum walks.

Usage

```
qwdap.swr(in_data, data_range, plotting)
```

Arguments

`in_data`	a 'QWMS' object, which includes the target series and the selected modes which can be obtained from modes selection.
`data_range`	the range of the train samples.
`plotting`	whether to plot.

Value

a 'QWMODEL' object which includes the information of regression analysis.

Examples

```
data("traffic.n1")
res.swr <- qwdap.swr(traffic.n1,c(1,500))
```

---

qwdap.sws *Mode Selection by Stepwise Regression*

Description

Mode selection by Stepwise Regression. The purpose of this function is to select the part modes with similar characteristics to the observed time series from the modes generated by the quantum walk. And it is based on the linear model. The core algorithm comes from StepReg(ver. 1.4.2).

Usage

```
qwdap.sws(real, ctqw, index, select_method, plotting)
```

Arguments

real	the real series observed.
ctqw	the 'CTQW' object.
index	the index of the data for mode selection.
select_method	choose a stepwise method.
plotting	whether to plot.

Details

The 'QWMS' object include the original time series and the modes generated by quantum walks.

Value

a 'QWMS' object.

Examples

```
data("traffic.qw")
data("trafficflow")
res.sws <- qwdap.sws(trafficflow,traffic.qw,1,"bidirection",TRUE)
```

qwdap.var — *Vector Autoregressive Model*

Description

Vector autoregressive model. This is a regression method used to establish the temporal relationship between the original time series and the modes generated by quantum walks. The core algorithm comes from MTS(ver. 1.1.1).

Usage

```
qwdap.var(in_data, data_range, plotting)
```

Arguments

`in_data`	a 'QWMS' object, which includes the target series and the selected modes which can be obtained from modes selection.
`data_range`	the range of the train samples.
`plotting`	whether to plot.

Value

a 'QWMODEL' object which includes the information of regression analysis.

Examples

```
data("traffic.n1")
res.var <- qwdap.var(traffic.n1,c(1,500))
```

traffic.model.n1 — *The estabulished model by Stepwise Regression of the 'N1' station*

Description

This data is the linear model built by Stepwise Regression of the highway traffic flow data of the 'N1' stations, and includes the observed data and the modes generated by quantum walk.

Usage

```
data(traffic.model.n1)
```

Format

A 'QWMODEL' object.

Source

Pan BH(2021).

`traffic.n1` *Data of the 'N1' station*

Description

This data is the highway traffic flow data and the modes generated by quantum walk of the 'N1' stations.

Usage

```
data(traffic.n1)
```

Format

A 'QWMS' object.

Source

Pan BH(2021).

`traffic.qw` *A set of modes generated by quantum walk*

Description

This data is generated by function 'qwdap.qwalk()' with 100 scaling factors form 0.01 to 1 and the parameter 'edges' is the adjancy matrix of 7 vertices connected end to end.

Usage

```
data(traffic.qw)
```

Format

A 'CTQW' object.

Source

Pan BH(2021).

`trafficflow` *Highway traffic flow data*

Description

This data set has a total of 720 records of 7 research stations, namely Tangshan (N1), Jurong (N2), Heyang (N3), Danyang (N4), Luoshuyan (N5), Xuejia (N6) and ChangzhouBei (N7).

Usage

```
data(trafficflow)
```

Format

A dataframe with 720 observations on the 7 stations.

Source

Yu ZY, Hu X(2020).

Index

∗ **datasets**
traffic.model.n1, 9
traffic.n1, 10
traffic.qw, 10
trafficflow, 11

qwdap.eval, 2
qwdap.pcr, 3
qwdap.plsr, 3
qwdap.ppr, 4
qwdap.predict, 5
qwdap.qwalk, 5
qwdap.rrelieff, 6
qwdap.swr, 7
qwdap.sws, 8
qwdap.var, 9

traffic.model.n1, 9
traffic.n1, 10
traffic.qw, 10
trafficflow, 11