Package ‘cutoffR’
August 29, 2016

Type Package
Title CUTOFF: A Spatio-temporal Imputation Method
Version 1.0
Date 2013-05-15
Author Lingbing Feng, Gen Nowak, Alan. H. Welsh, Terry. J. O’Neill
Maintainer Lingbing Feng <fenglb88@gmail.com>
Description This package provides a set of tools for spatio-temporal imputation in R. It includes the implementation for then CUTOFF imputation method, a useful cross-validation function that can be used not only by the CUTOFF method but also by some other imputation functions to help choosing an optimal value for relevant parameters, such as the number of k-nearest neighbors for the KNN imputation method, or the number of components for the SVD imputation method. It also contains tools for simulating data with missing values with respect to some specific missing pattern, for example, block missing. Some useful visualisation functions for imputation purposes are also provided in the package.
LazyLoad yes
Repository CRAN
Depends R (>= 3.1.0)
Imports ggplot2, reshape2
NeedsCompilation no
License GPL-2
Date/Publication 2014-05-13 07:44:40

R topics documented:

  complete.chunk ................................................. 2
  CosK ............................................................ 2
  Cut ............................................................... 3
  cutoff ........................................................... 4
date.month ......................................................... 5
### complete.chunk

*Complete Chunk Data* A chunk of data with no missing values from the Murray-Darling Basin Rainfall Data

### Description

- X020020. station No.020020
- X024501. station No.024501
- ...

### Format

A data frame with 576 rows and 78 variables

### CosK

*The Cosine Kernel*

### Description

The Cosine Kernel

### Usage

```r
CosK(x)
```

### Arguments

- `x` function arguments

### Examples

```r
curve(CosK)
plot(CosK, -2, 2)
```
The simple version of CUTOFF

Description
The simple version of CUTOFF

Usage
Cut(data, cutoff = 0.75, method = "pearson", ID = FALSE, ...)

Arguments
data a data matrix with missing values
cutoff the cutoff value for the CUTOFF method
method CUTOFF method to be used.
ID If the reference information needs to be retained during the imputation if TRUE, then reference information can be retained from the returned list by calling ID. If FALSE, then no reference information will be retained.
... other arguments

Value
If ID = FALSE, then return the imputed data matrix with no missing values. If ID = TRUE, then return a list of two components:

imputed The imputed data matrix with no missing values
ID The reference information during the imputation

References

Examples
data(hqmr.data)
#' # check the number of missing values
nmissing(hqmr.data[, -79])
# impute the data by the CUTOFF method
impdata <- Cut(data = hqmr.data)
nmissing(impdata)
The CUTOFF Spatio-temporal Imputation Method

Description

The CUTOFF Spatio-temporal Imputation Method

Usage

cutoff(data, N = 4, cutoff = 0.75, P = 5, M = floor(P/2), Adj = 1,
       space.weight = FALSE, method = c("correlation", "number", "penalty"),
       time.opts = c("average", "adjacent"), kernel = FALSE, kerFUN = NULL,
       lambda = NULL, corr = "pearson", keep.ID = FALSE, ...)

Arguments

data a matrix or data frame with missing values

N a number indicating the number used for the "CUTOFF by number" method

cutoff a number indicating the cutoff value used for the "CUTOFF by correlation" method

P a number for the "penalty" imputation option for CUTOFF. That is, for those candidate missing station with too many reference stations, we can penalise and fix the number of reference stations to P

M a number used for the "relaxation" imputation option for CUTOFF. That is, for those candidate missing station with too few reference stations, we can relax and add its number of reference stations to M

Adj a number used for the "adjacent" method in CUTOFF. That is, the missing value's adjacent points in time is also used for imputation. The default values is 1. 2 is also available. Any number bigger than 2 has not been implemented yet. This option is useful when the length of the time series is short so may be more temporal information can be useful to improve the imputation performance.

space.weight a logical value. If true, then space weighting strategy is carried out. The default is FALSE.

method the imputation method to be used. There are three options: "correlation", "number" and "penalty". Details can be found in Feng et al.(2014).

time.opts options for the temporal dimension; either "average" or "adjacent" can be used. "average" refers to simple averaging, "adjacent" refers to the "adjacent" method.

kernel logical, if TRUE then kernel smoothing can be used to smooth the averaging. Default is FALSE. If TRUE, then kerFUN has to be specified.

kerFUN the kernel function to be used for kernel smoothing. There are four kernel functions available in this package: EpanK, UnifK GaussK and CosK. User can define their own kernel function to pass to this function.

lambda a number indicating the bandwidth parameter value for kernel smoothing.
The type of correlation coefficient to be used for the "CUTOFF by correlation" method. Default is "pearson", "spearman" and "kendall" are alternatives.

If the reference ID for each missing stations need to be kept. If TRUE, relevant ID information can be retrieved after imputation. Default is FALSE.

other arguments that can passed

Details

This function implements the CUTOFF spatio-temporal imputation method that is described in Feng et al.(2014)

Value

If keep.ID = FALSE, then return the imputed data matrix with no missing values. If keep.ID = TRUE, then return a list of two components:

imputed The imputed data matrix with no missing values
ID The reference information during the imputation

References


Examples

```r
data(hqmr.data)
# check the number of missing values
nmissing(hqmr.data[, -79])
# impute the data by the CUTOFF method
impdata <- cutoff(data = hqmr.data)
nmissing(impdata)
```

Description

Date month data Date information for the Murray-Darling Basin rainfall data

Format

A vector of dates which length is 1200.
EpanK  

The Epanechnikov Kernel

Description

The Epanechnikov Kernel

Usage

EpanK(x)

Arguments

x  function arguments

Examples

curve(EpanK)
plot(EpanK, -2, 2)

GaussK  

The Gaussian Kernel

Description

The Gaussian Kernel

Usage

GaussK(x)

Arguments

x  function arguments

Examples

curve(GaussK)
plot(GaussK, -2, 2)
**Grmse**

*RMSE give imputed data matrix and the true matrix*

**Description**

RMSE give imputed data matrix and the true matrix

**Usage**

```r
 Grmse(ximp, xtrue)
```

**Arguments**

- `ximp` the imputed matrix
- `xtrue` the true matrix

**Value**

the RMSE

**Examples**

```r
 data(hqmr.data)
```

---

**HeatStruct**

*Structure Heatmap with Missing Value Demonstration*

**Description**

Structure Heatmap with Missing Value Demonstration

**Usage**

```r
 HeatStruct(data, high.col = "steelblue", low.col = "white",
           missing.col = "gold", xlab = "", ylab = "")
```

**Arguments**

- `data` a data frame or matrix, possibly with missing values denoted by NA
- `high.col` color for high values, can be a number or a color name, default is "steelblue"
- `low.col` color for high values, can be a number or a color name, default is "white"
- `missing.col` color for missing values, can be a number or a color name, default is "gold"
- `xlab` a title for the x axis.
- `ylab` a title for the y axis.
Details

Structure heatmap is like a normal heatmap, but is particularly useful when there are missing values in the data matrix. Default colors were carefully chosen so normally it is a good choice for your data. However, you are still encouraged to play around with it.

Examples

data(hqmr.data)
# use a subset of the hqmr.data
# notice the gold chunks which represent missing values
subdata <- hqmr.data[1000:1200, 1:30]
HeatStruct(subdata)
# change colors for high.col, low.col and missing.col
HeatStruct(subdata, low.col = "blue", high.col = "red", missing.col = "black")

Description

A dataset containing rainfall recordings from 78 gauging stations from the Murray-Darling Basin in Southeastern Australia.

Format

A data frame with 1200 rows and 79 variables

Details

- X020020. station No.020020
- X024501. station No.024501
- ...
- date.month month information

impCV

Cross-validation for spatio-temporal imputation

Description

Cross-validation for spatio-temporal imputation

Usage

impCV(data, FUN = Cut, date.info = TRUE, cfold = 10, rfold = 10, ...)
**MissSimulation**

**Arguments**

- `data`: a data matrix with missing values
- `FUN`: the imputation function to be evaluated by cross-validation
- `date.info`: logical, if date information is provided in the data.
- `cfold`: fold size on the columns
- `rfold`: fold size on the rows.
- `...`: other arguments

**Value**

the cross-validated RMSE

**Examples**

```r
data(hqmr.data)
# the real cross-validation will take some time to finish
# impCV(hqmr.data)
```

---

**MissSimulation**  
*Simulate a missing vector with block missing pattern.*

**Description**

Simulate a missing vector with block missing pattern.

**Usage**

```r
MissSimulation(n = 84, maxlen = 15, cnst = 15, prob = 0.03)
```

**Arguments**

- `n`: the length of the vector
- `maxlen`: the maximum length of missing
- `cnst`: the constant used to smooth the block missing
- `prob`: the probability a single element in the vector gets missing

**Value**

the same length vector with wanted block missing pattern
Examples

# default setting
rev1 <- MissSimulation()
# with larger missing probability
rev2 <- MissSimulation(prob = 0.5)
sum(is.na(rev1))
sum(is.na(rev2))

## Simulation block missing pattern in the Murray-Darling Basin rainfall data
BlockMissing <- function() {
  complete.chunk <- data(complete.chunk)
  block.size <- 3 # scale for blocks when simulating the first part
  n.years <- c(12, 36, 48, 48) # number of years for four simulation parts
  n.stations <- c(17, 17, 37, 24) # number of stations for four simulation parts
  n.prob <- c(0.05, 0.005, 0.005, 0.0005) # probability vector for each simulation part
  part1.sim <- function() MissSimulation(n = 4*n.years[1], maxlen=12, cnst=12, n.prob[1])
  part2.sim <- function() MissSimulation(n = 12*n.years[2], maxlen=3, cnst=3, n.prob[2])
  part3.sim <- function() MissSimulation(n = 12*n.years[3], maxlen=3, cnst=3, n.prob[3])
  part4.sim <- function() MissSimulation(n = 12*n.years[4], maxlen=3, cnst=3, n.prob[4])
  p1 <- function() {
    part1.mat <- matrix(0, nrow = 4*n.years[1], ncol = n.stations[1])
    for (j in 1:length(part1.mat[, 1])) {
      part1.mat[, j] <- part1.sim()
      part1.missing.mat <- matrix(0, nrow = 12*n.years[1], ncol = n.stations[1])
      # each block value should repeat three times to get the true missing matrix
      part1.missing.mat[, ] <- part1.mat[rep(1:nrow(part1.mat), each=block.size), ]
      part1.missing.mat[part1.missing.mat==1] <- NA
    }
    return(p1.miss = part1.missing.mat)
  }
  
  p2 <- function() {
    # simulate missing matrix part2
    part2.mat <- matrix(0, nrow=12*n.years[2], ncol=n.stations[2])
    for (j in 1:length(part2.mat[, 1])) {
      part2.mat[, j] <- part2.sim()
      part2.missing.mat <- part2.mat
      part2.missing.mat[part2.missing.mat==1] <- NA
    }
    return(p2.miss = part2.missing.mat)
  }
  
  p3 <- function() {
    # simulate missing matrix part3
    part3.mat <- matrix(0, nrow=12*n.years[3], ncol=n.stations[3])
    for (j in 1:length(part3.mat[, 1])) {
      part3.mat[, j] <- part3.sim()
      part3.missing.mat <- part3.mat
      part3.missing.mat[part3.missing.mat==1] <- NA
    }
    return(p3.miss = part3.missing.mat)
  }


```r
p4 <- function() {
  # simulate missing matrix part3
  part4.mat <- matrix(0, nrow=12*n.years[4], ncol=n.stations[4])
  for (j in 1:length(part4.mat[, 1])) {
    part4.mat[, j] <- part4.sim()
    part4.missing.mat <- part4.mat
    part4.missing.mat[part4.missing.mat==1] <- NA
  }
  return(part4.missing=part4.missing.mat)
}

return(complete.sim = as.data.frame(cbind(rbind(p2(), p1()), cbind(p3(), p4()))
          + complete.chunk)
)
# NOTRUN
# bdata <- BlockMissing()
# HeatStruct(bdata)
```

---

### nmissing

**Count the number of missing values in a vector or data matrix**

**Description**

Count the number of missing values in a vector or data matrix

**Usage**

```r
nmissing(x)
```

**Arguments**

- `x` a vector, matrix or data frame

**Value**

the number of missing values (denoted by NA)

**Examples**

```r
data(hqmr.data)
nmissing(hqmr.data)
```
The Uniform Kernel

**Description**

The Uniform Kernel

**Usage**

`UnifK(x)`

**Arguments**

- `x` function arguments

**Examples**

- `curve(UnifK)`
- `plot(UnifK, -2, 2)`
Index

*Topic datasets
   complete.chunk, 2
   date.month, 5
   hqmr.data, 8
   complete.chunk, 2
   CosK, 2
   Cut, 3
   cutoff, 4
   date.month, 5
   EpanK, 6
   GaussK, 6
   Grmse, 7
   HeatStruct, 7
   hqmr.data, 8
   impCV, 8
   MissSimulation, 9
   nmissing, 11
   UnifK, 12