Package 'DisaggregateTS'

January 20, 2025

Type Package

Title High-Dimensional Temporal Disaggregation

Version 3.0.1

Description Provides tools for temporal disaggregation, including:
(1) High-dimensional and low-dimensional series generation for simulation studies;
(2) A toolkit for temporal disaggregation and benchmarking using low-dimensional indicator series
as proposed by Dagum and Cholette (2006, ISBN:978-0-387-35439-2);
(3) Novel techniques by Mosley, Gibberd, and Eckley (2022, <doi:10.1111/rssa.12952>)
for disaggregating low-frequency series in the presence of high-dimensional indicator matrices.

License GPL-3

Encoding UTF-8

LazyData true

Imports Rdpack, stats, Matrix, lars, zoo, withr

Suggests testthat (>= 3.0.0), knitr, rmarkdown, readxl, corrplot, ggplot2

RdMacros Rdpack

Config/testthat/edition 3

RoxygenNote 7.2.3

VignetteBuilder knitr

Depends R (>= 3.5.0)

NeedsCompilation no

Author Kaveh Salehzadeh Nobari [aut, cre], Luke Mosley [aut]

Maintainer Kaveh Salehzadeh Nobari <k.salehzadeh-nobari@imperial.ac.uk>

Repository CRAN

Date/Publication 2024-10-31 12:40:06 UTC

Contents

`Data` *GHG Emissions and Financial Data for IBM*

Description

This dataset contains time series data on greenhouse gas (GHG) emissions and financial variables for IBM covering the period from Q3 2005 to Q3 2021. It is designed for use in demonstrating temporal disaggregation and adaptive LASSO methods for estimating high-frequency GHG emissions from low-frequency data.

Usage

```
Data
```

Format

A data frame with 68 rows (representing quarters) and 113 variables:

time Numeric vector representing the time index, spanning from Q3 2005 to Q3 2021

GHG Numeric vector of annual greenhouse gas emissions for IBM, recorded annually and repeated quarterly

financial_variables A matrix or data frame of 112 financial variables, extracted from quarterly balance sheets, income statements, and cash flow statements for each company

Source

Original data collected from financial statements and GHG reports of IBM.

`disaggregate` *Temporal Disaggregation Methods*

Description

This function contains the traditional standard-dimensional temporal disaggregation methods proposed by Denton (1971), Dagum and Cholette (2006), Chow and Lin (1971), Fernández (1981) and Litterman (1983), and the high-dimensional methods of Mosley et al. (2022).

Usage

```
disaggregate(
  Y,
  X = matrix(data = rep(1, times = (nrow(Y) * aggRatio)), nrow = (nrow(Y) * aggRatio)),
  aggMat = "sum",
  aggRatio = 4,
  method = "Chow-Lin",
  Denton = "additive-first-diff"
)
```

Arguments

`Y`	The low-frequency response series ($n_l \times 1$ matrix).
`X`	The high-frequency indicator series ($n \times p$ matrix).
`aggMat`	Aggregation matrix according to 'first', 'sum', 'average', 'last' (default is 'sum').
`aggRatio`	Aggregation ratio e.g. 4 for annual-to-quarterly, 3 for quarterly-to-monthly (default is 4).
`method`	Disaggregation method using 'Denton', 'Denton-Cholette', 'Chow-Lin', 'Fernandez', 'Litterman', 'spTD' or 'adaptive-spTD' (default is 'Chow-Lin').
`Denton`	Type of differencing for Denton method: 'simple-diff', 'additive-first-diff', 'additive-second-diff', 'proportional-first-diff' and 'proportional-second-diff' (default is 'additive-first-diff'). For instance, 'simple-diff' differencing refers to the differences between the original and revised values, whereas 'additive-first-diff' differencing refers to the differences between the first differenced original and revised values.

Details

Takes in a $n_l \times 1$ low-frequency series to be disaggregated Y and a $n \times p$ high-frequency matrix of p indicator series X. If $n > n_l \times aggRatio$ where $aggRatio$ is the aggregation ratio (e.g. $aggRatio = 4$ if annual-to-quarterly disagg, or $aggRatio = 3$ if quarterly-to-monthly disagg) then extrapolation is done to extrapolate up to n.

Value

`y_Est`: Estimated high-frequency response series (output is an $n \times 1$ matrix).

`beta_Est`: Estimated coefficient vector (output is a $p \times 1$ matrix).

`rho_Est`: Estimated residual AR(1) autocorrelation parameter.

`ul_Est`: Estimated aggregate residual series (output is an $n_l \times 1$ matrix).

References

Chow GC, Lin A (1971). "Best Linear Unbiased Interpolation, Distribution, and Extrapolation of Time Series by Related Series." *The review of Economics and Statistics*, **53**(4), 372–375.

Dagum EB, Cholette PA (2006). *Benchmarking, Temporal Distribution, and Reconciliation Methods for Time Series*. Springer.

Denton FT (1971). "Adjustment of monthly or quarterly series to annual totals: an approach based on quadratic minimization." *Journal of the american statistical association*, **66**(333), 99–102.

Fernández RB (1981). "A methodological note on the estimation of time series." *The Review of Economics and Statistics*, **63**(3), 471–476.

Litterman RB (1983). "A random walk, Markov model for the distribution of time series." *Journal of Business & Economic Statistics*, **1**(2), 169–173.

Mosley L, Eckley IA, Gibberd A (2022). "Sparse Temporal Disaggregation." *Journal of the Royal Statistical Society Series A: Statistics in Society*, **185**(4), 2203-2233. ISSN 0964-1998, doi:10.1111/rssa.12952, https://academic.oup.com/jrsssa/article-pdf/185/4/2203/49420183/jrsssa_185_4_2203.pdf.

Examples

```
data <- TempDisaggDGP(n_l=25,n=100,p=10,rho=0.5)
X <- data$X_Gen
Y <- data$Y_Gen
fit_chowlin <- disaggregate(Y=Y,X=X,method='Chow-Lin')
y_hat = fit_chowlin$y_Est
```

`simulDiagnosis` *Simulation Diagnostics*

Description

This function provides diagnostics for evaluating the accuracy of simulated data. Specifically, it computes the Mean Squared Error (MSE) between the true and estimated response vectors, and optionally, the sign recovery percentage of the coefficient vector.

Usage

```
simulDiagnosis(data_Hat, data_True, sgn = FALSE)
```

Arguments

data_Hat	List containing the estimated high-frequency data, with components y_Est (estimated response vector) and beta_Est (estimated coefficient vector).
data_True	List containing the true high-frequency data, with components y_Gen (true response vector) and Beta_Gen (true coefficient vector).
sgn	Logical value indicating whether to compute the sign recovery percentage. Default is FALSE.

Details

The function takes in the generated high-frequency data (data_True) and the estimated high-frequency data (data_Hat), and returns the Mean Squared Error (MSE) between the true and estimated values of the response vector. If the sgn parameter is set to TRUE, the function additionally computes the percentage of correctly recovered signs of the coefficient vector.

Value

If sgn is FALSE, the function returns the Mean Squared Error (MSE) between the true and estimated response vectors. If sgn is TRUE, the function returns a list containing both the MSE and the sign recovery percentage.

Examples

```
true_data <- list(y_Gen = c(1, 2, 3), Beta_Gen = c(1, -1, 0))
est_data <- list(y_Est = c(1.1, 1.9, 2.8), beta_Est = c(1, 1, 0))
mse <- simulDiagnosis(est_data, true_data)
results <- simulDiagnosis(est_data, true_data, sgn = TRUE)
```

TempDisaggDGP *High and Low-Frequency Data Generating Processes*

Description

This function generates a high-frequency response vector y, following the relationship $y = X\beta + \epsilon$, where X is a matrix of indicator series and β is a potentially sparse coefficient vector. The low-frequency vector Y is generated by aggregating y according to a specified aggregation method.

Usage

```
TempDisaggDGP(
  n_l,
  n,
  aggRatio = 4,
  p = 1,
  beta = 1,
  sparsity = 1,
```

```
  method = "Chow-Lin",
  aggMat = "sum",
  rho = 0,
  mean_X = 0,
  sd_X = 1,
  sd_e = 1,
  simul = FALSE,
  sparse_option = "random",
  setSeed = 42
)
```

Arguments

n_l	Integer. Size of the low-frequency series.
n	Integer. Size of the high-frequency series.
aggRatio	Integer. Aggregation ratio between low and high frequency (default is 4).
p	Integer. Number of high-frequency indicator series to include.
beta	Numeric. Value for the positive and negative elements of the coefficient vector.
sparsity	Numeric. Sparsity percentage of the coefficient vector (value between 0 and 1).
method	Character. The DGP of residuals to use ('Denton', 'Denton-Cholette', 'Chow-Lin', 'Fernandez', 'Litterman').
aggMat	Character. Aggregation matrix type ('first', 'sum', 'average', 'last').
rho	Numeric. Residual autocorrelation coefficient (default is 0).
mean_X	Numeric. Mean of the design matrix (default is 0).
sd_X	Numeric. Standard deviation of the design matrix (default is 1).
sd_e	Numeric. Standard deviation of the errors (default is 1).
simul	Logical. If TRUE, the design matrix and the coefficient vector are fixed (default is FALSE).
sparse_option	Character or Integer. Option to specify sparsity in the coefficient vector ('random' or integer value). Default is "random".
setSeed	Integer. Seed value for reproducibility when simul is set to TRUE.

Details

The aggregation ratio (aggRatio) determines the ratio between the low and high-frequency series (e.g., aggRatio = 4 for annual-to-quarterly). If the number of observations n exceeds $aggRatio \times n_l$, the aggregation matrix will include zero columns for the extrapolated values.

The function supports several data generating processes (DGP) for the residuals, including 'Denton', 'Denton-Cholette', 'Chow-Lin', 'Fernandez', and 'Litterman'. These methods differ in how they generate the high-frequency data and residuals, with optional autocorrelation specified by rho.

Value

A list containing the following components:

- `y_Gen`: Generated high-frequency response series (an $n \times 1$ matrix).
- `Y_Gen`: Generated low-frequency response series (an $n_l \times 1$ matrix).
- `X_Gen`: Generated high-frequency indicator series (an $n \times p$ matrix).
- `Beta_Gen`: Generated coefficient vector (a $p \times 1$ matrix).
- `e_Gen`: Generated high-frequency residual series (an $n \times 1$ matrix).

Examples

```
data <- TempDisaggDGP(n_l=25,n=100,p=10,rho=0.5)
X <- data$X_Gen
Y <- data$Y_Gen
```

Index