Package 'FoReco'

October 18, 2020

Type Package

Title Point Forecast Reconciliation

Version 0.1.1

Description Provides classical (bottom-up), optimal and heuristic combination forecast reconciliation procedures for cross-sectional, temporal, and cross-temporal linearly constrained time series.

License GPL-3

URL https://github.com/daniGiro/FoReco, https://danigiro.github.io/FoReco/

BugReports https://github.com/daniGiro/FoReco/issues

Depends R (>= 2.10), Matrix, osqp, stats

Imports cli, corpcor, methods, pracma

Suggests knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

NeedsCompilation no

RoxygenNote 7.1.1

Author Daniele Girolimetto [aut, cre, fnd], Tommaso Di Fonzo [aut, fnd]

Maintainer Daniele Girolimetto <daniele.girolimetto@phd.unipd.it>

Repository CRAN

Date/Publication 2020-10-17 22:20:03 UTC

R topics documented:

`FoReco-package` *FoReco: point forecast reconciliation*

Description

An R package offering classical (bottom-up), optimal and heuristic combination forecast reconciliation procedures for cross-sectional, temporal, and cross-temporal linearly constrained time series.

Details

The `FoReco` package is designed for point forecast reconciliation, a post-forecasting process aimed to improve the quality of the base forecasts for a system of linearly constrained (e.g. hierarchical/grouped) time series. The main functions are:

`htsrec()`: cross-sectional (contemporaneous) forecast reconciliation.

`thfrec()`: forecast reconciliation for a single time series through temporal hierarchies.

`tcsrec()`: heuristic first-temporal-then-cross-sectional cross-temporal forecast reconciliation.

`cstrec()`: heuristic first-cross-sectional-then-temporal cross-temporal forecast reconciliation.

`iterec()`: heuristic iterative cross-temporal forecast reconciliation.

`octrec()`: optimal combination cross-temporal forecast reconciliation.

Author(s)

Tommaso Di Fonzo and Daniele Girolimetto, Department of Statistical Sciences, University of Padua (Italy).

References

Di Fonzo, T., Girolimetto, D. (2020), Cross-Temporal Forecast Reconciliation: Optimal Combination Method and Heuristic Alternatives, Department of Statistical Sciences, University of Padua, arXiv:2006.08570.

Cmatrix *Cross-sectional (contemporaneous) aggregation matrix*

Description

This function allows the user to easily build the (na x nb) cross-sectional (contemporaneous) matrix mapping the nb bottom level series into the na higher level ones. (Experimental version)

Usage

```
Cmatrix(formula, data, sep = "_", sparse = TRUE, top_label = "Total")
```

Arguments

formula	Specification of the hierarchical structure: grouped hierarchies are specified using ~ g1 * g2 and nested hierarchies are specified using ~ parent / child. Mixtures of the two formulations are also possible, like ~ g1 * (grandparent / parent / child).
data	A dataset in which each column contains the values of the variables in the formula and each row identifies a bottom level time series.
sep	Character to separate the names of the aggregated series (*default* is "_").
sparse	Option to return sparse matrix (*default* is TRUE).
top_label	Label of the top level variable (*default* is "Total").

Value

A (na x nb) matrix.

Examples

```
## Balanced hierarchy
#         T
#    |--------|
#    A        B
#  |---|   |--|--|
# AA   AB  BA BB BC
# Names of the bottom level variables
data_bts <- data.frame(X1 = c("A", "A", "B", "B", "B"),
                       X2 = c("A", "B", "A", "B", "C"),
                       stringsAsFactors = FALSE)
# Cross-sectional aggregation matrix
C <- Cmatrix(~ X1 / X2, data_bts, sep = "")
```

```
## Unbalanced hierarchy (1)
#             T
#    |--------|------|
#    A        B      C
#  |---|   |--|--|
# AA   AB  BA BB BC
# Names of the bottom level variables
data_bts <- data.frame(X1 = c("A", "A", "B", "B", "B", "C"),
                       X2 = c("A", "B", "A", "B", "C", NA),
                       stringsAsFactors = FALSE)
# Cross-sectional aggregation matrix
C <- Cmatrix(~ X1 / X2, data_bts, sep = "")

## Unbalanced hierarchy (2)
#                 T
#       |---------|---------|
#       A         B         C
#     |---|     |---|     |---|
#    AA   AB   BA   BB   CA   CB
#  |----|         |----|
# AAA  AAB       BBA  BBB
# Names of the bottom level variables
data_bts <- data.frame(X1 = c("A", "A", "A", "B", "B", "B", "C", "C"),
                       X2 = c("A", "A", "B", "A", "B", "B", "A", "B"),
                       X3 = c("A", "B", NA, NA, "A", "B", NA, NA),
                       stringsAsFactors = FALSE)
# Cross-sectional aggregation matrix
C <- Cmatrix(~ X1 / X2 / X3, data_bts, sep = "")

## Grouped hierarchy
#         C               S
#    |--------|      |--------|
#    A        B      M        F
#  |---|    |---|
# AA   AB  BA   BB
# Names of the bottom level variables
data_bts <- data.frame(X1 = c("A", "A", "B", "B", "A", "A", "B", "B"),
                       X2 = c("A", "B", "A", "B", "A", "B", "A", "B"),
                       Y1 = c("M", "M", "M", "M", "F", "F", "F", "F"),
                       stringsAsFactors = FALSE)
# Cross-sectional aggregation matrix
C <- Cmatrix(~ Y1 * (X1 / X2), data_bts, sep = "")
```

---

commat *Commutation matrix*

Description

This function returns the ((r c) x (r c)) commutation matrix P such that

$$Pvec(Y) = vec(Y'),$$

where Y is a (r x c) matrix.

Usage

```
commat(r, c)
```

Arguments

r	Number of rows of Y.
c	Number of columns of Y.

Value

A sparse ((r c) x (r c)) matrix.

References

Magnus, J.R., Neudecker, H. (2019), Matrix Differential Calculus with Applications in Statistics and Econometrics, third edition, New York, Wiley, pp. 54-55.

Examples

```
Y <- matrix(rnorm(30), 5, 6)
P <- commat(5, 6)
P %*% as.vector(Y) == as.vector(t(Y)) # check
```

---

cstrec *Heuristic first-cross-sectional-then-temporal cross-temporal forecast reconciliation*

Description

The order of application of the two reconciliation steps proposed by Kourentzes and Athanasopoulos (2019), implemented in the function tcsrec, may be inverted. The function cstrec performs cross-sectional reconciliation (htsrec) first, then temporal reconciliation (thfrec), and finally applies the average of the projection matrices obtained in the second step to the one dimensional reconciled values obtained in the first step.

Usage

```
cstrec(basef, m, C, thf_comb, hts_comb, Ut, nb, res, W, Omega,
       mse = TRUE, corpcor = FALSE, nn = FALSE,
       settings = osqpSettings(verbose = FALSE, eps_abs = 1e-5,
       eps_rel = 1e-5, polish_refine_iter = 100, polish = TRUE))
```

Arguments

basef
: (n x h(k* + m)) matrix of base forecasts to be reconciled; n is the total number of variables, m is the highest frequency, k* is the sum of (p-1) factors of m, excluding m, and h is the forecast horizon. Each row identifies, a time series, and the forecasts are ordered as [lowest_freq' ... highest_freq']'.

m
: Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation).

C
: (na x nb) cross-sectional (contemporaneous) matrix mapping the bottom level series into the higher level ones.

thf_comb
: Type of the ((k* + m) x (k* + m)) covariance matrix to be used in the temporal reconciliation, see more in comb param of thfrec.

hts_comb
: Type of the (n x n) covariance matrix to be used in the cross-sectional reconciliation, see more in comb param of htsrec.

Ut
: Zero constraints cross-sectional (contemporaneous) kernel matrix ($\mathbf{U}'\mathbf{Y} = \mathbf{0}$) spanning the null space valid for the reconciled forecasts. It can be used instead of parameter C, but in this case nb (n = na + nb) is needed. If the hierarchy admits a structural representation, Ut has dimension (na x n).

nb
: Number of bottom time series; if C is present, nb is not used.

res
: (n x N(k* + m)) matrix containing the residuals at all the temporal frequencies, ordered [lowest_freq' ... highest_freq']' (columns) for each variable (row), needed to estimate the covariance matrix when hts_comb = {"wls", "shr", "sam"} and/or hts_comb = {"wlsv", "wlsh", "acov", "strar1", "sar1", "har1", "shr", "sam"}. The rows must be in the same order as basef.

W
: This option permits to directly enter the covariance matrix in the cross-sectional reconciliation, see more in W param of htsrec.

Omega
: This option permits to directly enter the covariance matrix in the reconciliation through temporal hierarchies, see more in Omega param of thfrec.

mse
: Logical value: TRUE (*default*) calculates the covariance matrix of the in-sample residuals (when necessary) according to the original **hts** and **thief** formulation: no mean correction, T as denominator.

corpcor
: Logical value: TRUE if **corpcor** (Schäfer et al., 2017) must be used to shrink the sample covariance matrix according to Schäfer and Strimmer (2005), otherwise the function uses the same implementation as package **hts**.

nn
: Logical value, TRUE if non-negative reconciled forecasts are wished. **Warning**, the two-step heuristic reconciliation allows non negativity constraints only in the first step. This means that non-negativity is not guaranteed in the final reconciled values.

settings
: Settings for **osqp** (object osqpSettings). The default options are: verbose = FALSE, eps_abs = 1e-5, eps_rel = 1e-5, polish_refine_iter = 100 and polish = TRUE. For details, see the **osqp** documentation (Stellato et al., 2019).

Value

The function returns a list with two elements:

recf	(n x h(k* + m)) reconciled forecasts matrix.
M	Projection matrix (projection approach).

References

Di Fonzo, T., Girolimetto, D. (2020), Cross-Temporal Forecast Reconciliation: Optimal Combination Method and Heuristic Alternatives, Department of Statistical Sciences, University of Padua, arXiv:2006.08570.

Kourentzes, N., Athanasopoulos, G. (2019), Cross-temporal coherent forecasts for Australian tourism, *Annals of Tourism Research*, 75, 393-409.

Schäfer, J.L., Opgen-Rhein, R., Zuber, V., Ahdesmaki, M., Duarte Silva, A.P., Strimmer, K. (2017), *Package 'corpcor'*, R package version 1.6.9 (April 1, 2017), https://CRAN.R-project.org/package=corpcor.

Schäfer, J.L., Strimmer, K. (2005), A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics, *Statistical Applications in Genetics and Molecular Biology*, 4, 1.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S. (2018). OSQP: An Operator Splitting Solver for Quadratic Programs, arXiv:1711.08013.

Stellato, B., Banjac, G., Goulart, P., Boyd, S., Anderson, E. (2019), OSQP: Quadratic Programming Solver using the 'OSQP' Library, R package version 0.6.0.3 (October 10, 2019), https://CRAN.R-project.org/package=osqp.

Examples

```
data(FoReco_data)
obj <- cstrec(FoReco_data$base, m = 12, C = FoReco_data$C, thf_comb = "acov",
              hts_comb = "shr", res = FoReco_data$res)
```

---

ctbu *Bottom-up Cross-temporal forecast reconciliation*

Description

Cross temporal reconciled forecasts for all series at any temporal aggregation level can be easily computed by appropriate summation of the high-frequency bottom base forecasts $\hat{\mathbf{b}}_i, i = 1, ..., n_b$, according to a bottom-up procedure like what is currently done in both the cross-sectional and temporal frameworks.

Usage

```
ctbu(Bmat, m, C)
```

Arguments

Bmat	(nb x (h m)) matrix of high-frequency bottom time series base forecasts. h is the forecast horizon for the lowest frequency (most temporally aggregated) time series.
m	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation).
C	(na x nb) cross-sectional (contemporaneous) matrix mapping the bottom level series into the higher level ones.

Value

The function returns a (n x (h (k* + m))) matrix of bottom-up cross-temporally reconciled forecasts.

References

Di Fonzo, T., Girolimetto, D. (2020), Cross-Temporal Forecast Reconciliation: Optimal Combination Method and Heuristic Alternatives, Department of Statistical Sciences, University of Padua, arXiv:2006.08570.

Examples

```
data(FoReco_data)
id <- which(simplify2array(strsplit(colnames(FoReco_data$base),
                                    split = "_"))[1, ] == "k1")
hfbts <- FoReco_data$base[-c(1:3), id]
obj <- ctbu(Bmat = hfbts, m = 12, C = FoReco_data$C)
rownames(obj) <- rownames(FoReco_data$base)
```

---

ctf_tools — *Cross-temporal reconciliation tools*

Description

Some useful tools for the cross-temporal forecast reconciliation of cross-sectionally and temporally constrained time series

Usage

```
ctf_tools(C, m, h = 1, Ut, nb, Sstruc = FALSE)
```

Arguments

- `C` — (na x nb) cross-sectional (contemporaneous) matrix mapping the bottom level series into the higher level ones.
- `m` — Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation).
- `h` — Forecast horizon for the lowest frequency (most temporally aggregated) time series (*default* is 1).
- `Ut` — Zero constraints cross-sectional (contemporaneous) kernel matrix $(\mathbf{U}'\mathbf{Y} = \mathbf{0})$ spanning the null space valid for the reconciled forecasts. It can be used instead of parameter `C`, but in this case `nb` (n = na + nb) is needed. If the hierarchy admits a structural representation, `Ut` has dimension (na x n).
- `nb` — Number of bottom time series; if `C` is present, `nb` is not used.
- `Sstruc` — If `Sstruc` = `TRUE` the function returns also the structural representation matrix of a cross-temporal system (*default* is `FALSE`).

Value

ctf list with:

- `Ht` — Full row-rank cross-temporal zero-valued constraints (kernel) matrix, $\mathbf{H}'\mathbf{y} = \mathbf{0}$.
- `Htbreve` — Complete, not full row-rank cross-temporal zero-valued constraints (kernel) matrix.
- `Htstruc` — Zero constraints full row-rank cross-temporal kernel matrix (structural representation) , $\check{\mathbf{H}}'$.
- `Sstruc` — Cross-temporal summing matrix (structural representation), $\check{\mathbf{S}}$.

hts list from `hts_tools`

thf list from `thf_tools`

Examples

```
# One level hierarchy (na = 1, nb = 2) with quarterly data
obj <- ctf_tools(C = matrix(c(1, 1), 1), m = 4, Sstruc = TRUE)
```

`FoReco-hts` *Simple examples to compare* `FoReco` *and* `hts` *packages*

Description

Two datasets of the **hts** package are used to show how to get the same results using **FoReco**. First, we consider the `htseg1` dataset (a simulated three level hierarchy, with a total of 8 series, each of length 10). Then, we take the `htseg2` dataset (a simulated four level hierarchy with a total of 17 series, each of length 16). `htseg1` and `htseg2` are objects of class `hts` in **hts**.

References

Hyndman, R. J., Lee, A., Wang, E., and Wickramasuriya, S. (2020). hts: Hierarchical and Grouped Time Series, *R package version 6.0.1*, https://CRAN.R-project.org/package=hts.

Examples

```
## Not run:
library(hts)
require(FoReco)

####### htseg1 #######
data <- allts(htseg1)
n <- NCOL(data)
nb <- NCOL(htseg1$bts)
na <- n-nb
C <- smatrix(htseg1)[1:na, ]

# List containing the base forecasts
# Forecast horizon: 10
base <- list()
for (i in 1:n) {
  base[[i]] <- forecast(auto.arima(data[, i]))
}

# Create the matrix of base forecasts
BASE <- NULL
for (i in 1:n) {
  BASE <- cbind(BASE, base[[i]]$mean)
}
colnames(BASE) <- colnames(data)

# Create the matrix of residuals
res <- NULL
for (i in 1:n) {
  res <- cbind(res, base[[i]]$residuals)
}
colnames(res) <- colnames(data)

# ols
Y_hts_forecast <- forecast(htseg1, method = "comb", fmethod = "arima", weights = "ols")
Y_hts_ols <- combinef(BASE, nodes = get_nodes(htseg1), keep = "all")
sum((allts(Y_hts_forecast) - Y_hts_ols) > 1e-10)
Y_FoReco_ols <- htsrec(BASE, C = C, comb = "ols")$recf
sum((Y_hts_ols - Y_FoReco_ols) > 1e-10)

# struc
w <- 1 / apply(smatrix(htseg1), 1, sum)
Y_hts_struc <- combinef(BASE, nodes = get_nodes(htseg1), weights = w, keep = "all")
Y_FoReco_struc <- htsrec(BASE, C = C, comb = "struc")$recf
sum((Y_hts_struc - Y_FoReco_struc) > 1e-10)

# shr
```

```
Y_hts_shr <- MinT(BASE, nodes = get_nodes(htseg1), keep = "all", covariance = "shr", residual = res)
Y_FoReco_shr <- htsrec(BASE, C = C, comb = "shr", res = res)$recf
sum((Y_hts_shr - Y_FoReco_shr) > 1e-10)

# sam - hts error "MinT needs covariance matrix to be positive definite"
#       The covariance matrix is non-singular, but its condition number is very low,
#       and hts considers it as non-invertible
Y_hts_sam <- MinT(BASE, nodes = get_nodes(htseg1), keep = "all", covariance = "sam", residual = res)
Y_FoReco_sam <- htsrec(BASE, C = C, comb = "sam", res = res)$recf
# sum((Y_hts_sam-Y_FoReco_sam)>1e-10)

####### htseg2 #######
data <- allts(htseg2)
n <- NCOL(data)
nb <- NCOL(htseg2$bts)
na <- n-nb
C <- smatrix(htseg2)[1:na, ]

## In FoReco, forecasts must be obtained externally
# List containing the base forecasts
# Forecast horizon: 10
base <- list()
for (i in 1:n) {
  base[[i]] <- forecast(auto.arima(data[, i]))
}

# Create the matrix of base forecasts
BASE <- NULL
for (i in 1:n) {
  BASE <- cbind(BASE, base[[i]]$mean)
}
colnames(BASE) <- colnames(data)

# Create the matrix of residuals
res <- NULL
for (i in 1:n) {
  res <- cbind(res, base[[i]]$residuals)
}
colnames(res) <- colnames(data)

## Comparison
# ols
Y_hts_forecast <- forecast(htseg2, method = "comb", fmethod = "arima", weights = "ols")
Y_hts_ols <- combinef(BASE, nodes = get_nodes(htseg2), keep = "all")
sum((allts(Y_hts_forecast) - Y_hts_ols) > 1e-10)
Y_FoReco_ols <- htsrec(BASE, C = C, comb = "ols")$recf
sum(abs(Y_hts_ols - Y_FoReco_ols) > 1e-10)

# struc
w <- 1 / apply(smatrix(htseg2), 1, sum)
Y_hts_struc <- combinef(BASE, nodes = get_nodes(htseg2), weights = w, keep = "all")
Y_FoReco_struc <- htsrec(BASE, C = C, comb = "struc")$recf
sum(abs(Y_hts_struc - Y_FoReco_struc) > 1e-10)
```

```
# shr
Y_hts_shr <- MinT(BASE, nodes = get_nodes(htseg2), keep = "all", covariance = "shr", residual = res)
Y_FoReco_shr <- htsrec(BASE, C = C, comb = "shr", res = res)$recf
sum(abs(Y_hts_shr- Y_FoReco_shr) > 1e-10)

## End(Not run)
```

FoReco-thief *Simple examples to compare* FoReco *and* thief *packages*

Description

The dataset in the thief package is used to show how to get the same results with the FoReco package. In particular, we take the weekly data of Accident and Emergency demand in the UK, AEdemand, from 1 January 2011 to 31 December 2014.

References

Hyndman, R. J., Kourentzes, N. (2018), thief: Temporal HIErarchical Forecasting, *R package version 0.3*, https://cran.r-project.org/package=thief.

Examples

```
## Not run:
library(thief)
require(FoReco)
dataset <- window(AEdemand[, 12], start = c(2011, 1), end = c(2014, 52))
data <- tsaggregates(dataset)
# Base forecasts
base <- list()
for (i in 1:5) {
  base[[i]] <- forecast(auto.arima(data[[i]]))
}
base[[6]] <- forecast(auto.arima(data[[6]]), h = 2)
# Base forecasts vector
base_vec <- NULL
for (i in 6:1) {
  base_vec <- c(base_vec, base[[i]]$mean)
}
# Residual vector
res <- NULL
for (i in 6:1) {
  res <- c(res, base[[i]]$residuals)
}

# OLS
obj_thief <- thief(dataset, m = 52, h = 2 * 52, comb = "ols", usemodel = "arima")
obj_thief <- tsaggregates(obj_thief$mean)
y_thief <- NULL
```

```
for (i in 6:1) {
  y_thief <- c(y_thief, obj_thief[[i]])
}
obj_thief_ols <- reconcilethief(base, comb="ols")
y_thief_ols <- NULL
for (i in 6:1) {
  y_thief_ols <- c(y_thief_ols, obj_thief_ols[[i]]$mean)
}
sum(abs(y_thief_ols - y_thief) > 1e-10)
y_FoReco_ols <- thfrec(base_vec, 52, comb = "ols")$recf
sum(abs(y_FoReco_ols - y_thief_ols) > 1e-10)

# STRUC
obj_thief_struc <- reconcilethief(base, comb="struc")
y_thief_struc <- NULL
for (i in 6:1) {
  y_thief_struc <- c(y_thief_struc, obj_thief_struc[[i]]$mean)
}
y_FoReco_struc <- thfrec(base_vec, 52, comb = "struc")$recf
sum(abs(y_FoReco_struc - y_thief_struc) > 1e-10)

# BU
obj_thief_bu <- reconcilethief(base, comb="bu")
y_thief_bu <- NULL
for (i in 6:1) {
  y_thief_bu <- c(y_thief_bu, obj_thief_bu[[i]]$mean)
}
y_FoReco_bu <- thfrec(base_vec, 52, comb = "bu")$recf
sum(abs(y_FoReco_bu - y_thief_bu) > 1e-10)

# SHR
obj_thief_shr <- reconcilethief(base, comb="shr")
y_thief_shr <- NULL
for (i in 6:1) {
  y_thief_shr <- c(y_thief_shr, obj_thief_shr[[i]]$mean)
}
y_FoReco_shr <- thfrec(base_vec, 52, comb = "shr", res = res)$recf
sum(abs(y_FoReco_shr - y_thief_shr) > 1e-10)

## End(Not run)
```

`FoReco2ts` *Reconciled forecasts matrix/vector to time-series class*

Description

Function to transform the matrix/vector of reconciled forecasts (`recf` from htsrec, thfrec, octrec, tcsrec, cstrec, iterec, ctbu) into a list of time series objects.

Usage

```
FoReco2ts(recf, ...)
```

Arguments

`recf`	(h(k* + m) x 1) reconciled forecasts vector from thfrec, (h x n) reconciled forecasts matrix from htsrec or (n x h(k* + m)) reconciled forecasts matrix from octrec, tcsrec, cstrec, iterec, ctbu.
`...`	optional arguments to ts (i.e. starting date); frequency is only required for the cross-sectional system. .

Value

A list of class "ts" objects

Examples

```
data(FoReco_data)
# Cross-temporal framework
oct_recf <- octrec(FoReco_data$base, m = 12, C = FoReco_data$C,
                   comb = "bdshr", res = FoReco_data$res)$recf
obj_oct <- FoReco2ts(recf = oct_recf, start = c(15, 1))

# Cross-sectional framework
# monthly base forecasts
id <- which(simplify2array(strsplit(colnames(FoReco_data$base),
                                    split = "_"))[1, ] == "k1")
mbase <- t(FoReco_data$base[, id])
# monthly residuals
id <- which(simplify2array(strsplit(colnames(FoReco_data$res),
                                    split = "_"))[1, ] == "k1")
mres <- t(FoReco_data$res[, id])
hts_recf <- htsrec(mbase, C = FoReco_data$C, comb = "shr", res = mres)$recf
obj_hts <- FoReco2ts(recf = hts_recf, start = c(15, 1), frequency = 12)

# Temporal framework
# top ts base forecasts ([lowest_freq' ...  highest_freq']')
topbase <- FoReco_data$base[1, ]
 # top ts residuals ([lowest_freq' ...  highest_freq']')
topres <- FoReco_data$res[1, ]
thf_recf <- thfrec(topbase, m = 12, comb = "acov", res = topres)$recf
obj_thf <- FoReco2ts(recf = thf_recf, start = c(15, 1))
```

`FoReco_data` *Forecast reconciliation for a simulated hierarchical time series*

Description

A two-level hierarchy with n = 8 monthly time series. In the cross-sectional framework, at any time it is Tot = A + B + C, A = AA + AB and B = BA + BB (nb = 5 at the bottom level). For monthly data, the observations are aggregated to annual (k = 12), semi-annual (k = 6), four-monthly (k = 4), quarterly (k = 3), and bi-monthly (k = 2) observations. The monthly bottom time series are simulated from five different SARIMA models (see Using the ‘FoReco‘ package). There are 180 (15 years) monthly observations: the first 168 values (14 years) are used as training set, and the last 12 form the test set.

Usage

```
data(FoReco_data)
```

Format

An object of class "list":

base (8 x 28) matrix of base forecasts. Each row identifies a time series and the forecasts are ordered as [lowest_freq' ... highest_freq']'.

test (8 x 28) matrix of test set. Each row identifies a time series and the observed values are ordered as [lowest_freq' ... highest_freq']'.

res (8 x 392) matrix of in-sample residuals. Each row identifies a time series and the in-sample residuals are ordered as [lowest_freq' ... highest_freq']'.

C (3 x 5) cross-sectional (contemporaneous) aggregation matrix.

obs List of the observations at any levels and temporal frequencies.

Examples

```
data(FoReco_data)
# Cross-sectional reconciliation for all temporal aggregation levels
# (monthly, bi-monthly, ..., annual)
K <- c(1,2,3,4,6,12)
hts_recf <- NULL
for(i in 1:length(K)){
  # base forecasts
  id <- which(simplify2array(strsplit(colnames(FoReco_data$base),
                                      split = "_"))[1, ] == paste("k", K[i], sep=""))
  mbase <- t(FoReco_data$base[, id])
  # residuals
  id <- which(simplify2array(strsplit(colnames(FoReco_data$res),
                                      split = "_"))[1, ] == "k1")
  mres <- t(FoReco_data$res[, id])
  hts_recf[[i]] <- htsrec(mbase, C = FoReco_data$C, comb = "shr",
                          res = mres, keep = "recf")
}
names(hts_recf) <- paste("k", K, sep="")

# Forecast reconciliation through temporal hierarchies for all time series
n <- NROW(FoReco_data$base)
```

```
thf_recf <- matrix(NA, n, NCOL(FoReco_data$base))
dimnames(thf_recf) <- dimnames(FoReco_data$base)
for(i in 1:n){
  # ts base forecasts ([lowest_freq' ...  highest_freq']')
  tsbase <- FoReco_data$base[i, ]
  # ts residuals ([lowest_freq' ...  highest_freq']')
  tsres <- FoReco_data$res[i, ]
  thf_recf[i,] <- thfrec(tsbase, m = 12, comb = "acov",
                         res = tsres, keep = "recf")
}

# Iterative cross-temporal reconciliation
ite_recf <- iterec(FoReco_data$base, note=FALSE,
                   m = 12, C = FoReco_data$C,
                   thf_comb = "acov", hts_comb = "shr",
                   res = FoReco_data$res, start_rec = "thf")$recf

# Heuristic first-cross-sectional-then-temporal cross-temporal reconciliation
cst_recf <- cstrec(FoReco_data$base, m = 12, C = FoReco_data$C,
                   thf_comb = "acov", hts_comb = "shr",
                   res = FoReco_data$res)$recf

# Heuristic first-temporal-then-cross-sectional cross-temporal reconciliation
tcs_recf <- tcsrec(FoReco_data$base, m = 12, C = FoReco_data$C,
                   thf_comb = "acov", hts_comb = "shr",
                   res = FoReco_data$res)$recf

# Optimal cross-temporal reconciliation
oct_recf <- octrec(FoReco_data$base, m = 12, C = FoReco_data$C,
                   comb = "bdshr", res = FoReco_data$res, keep = "recf")
```

---

htsrec *Cross-sectional (contemporaneous) forecast reconciliation*

Description

Cross-sectional (contemporaneous) forecast reconciliation of hierarchical and grouped time series. The reconciled forecasts are calculated either through a projection approach (Byron, 1978, see also van Erven and Cugliari, 2015, and Wickramasuriya et al., 2019), or the equivalent structural approach by Hyndman et al. (2011). Moreover, the classic bottom-up approach is available.

Usage

```
htsrec(basef, comb, C, res, Ut, nb, mse = TRUE, corpcor = FALSE, W,
       type = "M", sol = "direct", nn = FALSE, keep = "list",
       settings = osqpSettings(verbose = FALSE, eps_abs = 1e-5,
       eps_rel = 1e-5, polish_refine_iter = 100, polish=TRUE))
```

Arguments

basef
: (h x n) matrix of base forecasts to be reconciled; h is the forecast horizon and n is the total number of time series.

comb
: Type of the reconciliation. Except for Bottom-up, each option corresponds to a specified (n x n) covariance matrix:
  - **bu** (Bottom-up);
  - **ols** (Identity);
  - **struc** (Structural variances);
  - **wls** (Series variances) - uses res;
  - **shr** (Shrunk covariance matrix - MinT-shr) - uses res;
  - **sam** (Sample covariance matrix - MinT-sam) - uses res;
  - **w** use your personal matrix W in param W.

C
: (na x nb) cross-sectional (contemporaneous) matrix mapping the bottom level series into the higher level ones.

res
: (N x n) in-sample residuals matrix needed when comb = {"wls", "shr", "sam"}. The columns must be in the same order as basef.

Ut
: Zero constraints cross-sectional (contemporaneous) kernel matrix ($\mathbf{U}'\mathbf{Y} = \mathbf{0}$) spanning the null space valid for the reconciled forecasts. It can be used instead of parameter C, but in this case nb (n = na + nb) is needed. If the hierarchy admits a structural representation, Ut has dimension (na x n).

nb
: Number of bottom time series; if C is present, nb is not used.

mse
: Logical value: TRUE (*default*) calculates the covariance matrix of the in-sample residuals (when necessary) according to the original **hts** and **thief** formulation: no mean correction, T as denominator.

corpcor
: Logical value: TRUE if **corpcor** (Schäfer et al., 2017) must be used to shrink the sample covariance matrix according to Schäfer and Strimmer (2005), otherwise the function uses the same implementation as package **hts**.

W
: This option permits to directly enter the covariance matrix:
  1. W must be a p.d. (n x n) matrix;
  2. if comb is different from "w", W is not used.

type
: Approach used to compute the reconciled forecasts: "M" for the projection approach with matrix M (*default*), or "S" for the structural approach with summing matrix S.

sol
: Solution technique for the reconciliation problem: either "direct" (*default*) for the direct solution or "osqp" for the numerical solution (solving a linearly constrained quadratic program using solve_osqp).

nn
: Logical value: TRUE if non-negative reconciled forecasts are wished.

keep
: Return a list object of the reconciled forecasts at all levels.

settings
: Settings for **osqp** (object osqpSettings). The default options are: verbose = FALSE, eps_abs = 1e-5, eps_rel = 1e-5, polish_refine_iter = 100 and polish = TRUE. For details, see the **osqp** documentation (Stellato et al., 2019).

Details

In case of non-negativity constraints, there are two ways:

1. sol = "direct" and nn = TRUE: the base forecasts will be reconciled at first without non-negativity constraints, then, if negative reconciled values are present, the "osqp" solver is used.
2. sol = "osqp" and nn = TRUE: the base forecasts will be reconciled through the "osqp" solver.

Value

If the parameter keep is equal to "recf", then the function returns only the (h x n) reconciled forecasts matrix, otherwise (keep="all") it returns a list that mainly depends on what type of representation (type) and methodology (sol) have been used:

recf (h x n) reconciled forecasts matrix.

W Covariance matrix used for forecast reconciliation.

nn_check Number of negative values (if zero, there are no values below zero).

rec_check Logical value: has the hierarchy been respected?

M (type="M" and type="direct")
Projection matrix (projection approach)

G (type="S" and type="direct")
Projection matrix (structural approach).

S (type="S" and type="direct")
Cross-sectional summing matrix, **S**.

info (type="osqp")
matrix with information in columns for each forecast horizon h (rows): run time (run_time) number of iteration, norm of primal residual (pri_res), status of osqp's solution (status) and polish's status (status_polish).

Only if comb = "bu", the function returns recf, S and M.

References

Byron, R.P. (1978), The estimation of large social accounts matrices, *Journal of the Royal Statistical Society A*, 141, 3, 359-367.

Di Fonzo, T., Girolimetto, D. (2020), Cross-Temporal Forecast Reconciliation: Optimal Combination Method and Heuristic Alternatives, Department of Statistical Sciences, University of Padua, arXiv:2006.08570.

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G., Shang, H.L.(2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589.

Schäfer, J.L., Opgen-Rhein, R., Zuber, V., Ahdesmaki, M., Duarte Silva, A.P., Strimmer, K. (2017), *Package 'corpcor'*, R package version 1.6.9 (April 1, 2017), https://CRAN.R-project.org/package=corpcor.

Schäfer, J.L., Strimmer, K. (2005), A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics, *Statistical Applications in Genetics and Molecular Biology*, 4, 1.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S. (2018). OSQP: An Operator Splitting Solver for Quadratic Programs, arXiv:1711.08013.

Stellato, B., Banjac, G., Goulart, P., Boyd, S., Anderson, E. (2019), OSQP: Quadratic Programming Solver using the 'OSQP' Library, R package version 0.6.0.3 (October 10, 2019), https://CRAN.R-project.org/package=osqp.

van Erven, T., Cugliari, J. (2015), Game-theoretically Optimal Reconciliation of Contemporaneous Hierarchical Time Series Forecasts, in Antoniadis, A., Poggi, J.M., Brossat, X. (eds.), *Modeling and Stochastic Learning for Forecasting in High Dimensions*, Berlin, Springer, 297-317.

Wickramasuriya, S.L., Athanasopoulos, G., Hyndman, R.J. (2019), Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization, *Journal of the American Statistical Association*, 114, 526, 804-819.

Examples

```
data(FoReco_data)
# monthly base forecasts
id <- which(simplify2array(strsplit(colnames(FoReco_data$base),
                                    split = "_"))[1, ] == "k1")
mbase <- t(FoReco_data$base[, id])
# monthly residuals
id <- which(simplify2array(strsplit(colnames(FoReco_data$res),
                                    split = "_"))[1, ] == "k1")
mres <- t(FoReco_data$res[, id])
obj <- htsrec(mbase, C = FoReco_data$C, comb = "shr", res = mres)
```

---

hts_tools *Cross-sectional reconciliation tools*

Description

Some useful tools for the cross-sectional reconciliation of linearly and hierarchically constrained time series

Usage

```
hts_tools(C, h = 1, Ut, nb, sparse = TRUE)
```

Arguments

C	(na x nb) cross-sectional (contemporaneous) matrix mapping the bottom level series into the higher level ones.
h	Forecast horizon (*default* is 1).
Ut	(na x n) zero constraints cross-sectional (contemporaneous) kernel matrix $\mathbf{U}'\mathbf{Y} = \mathbf{0}_{[n_a \times (k^* + m)]}$ spanning the null space valid for the reconciled forecasts. It can be used instead of parameter C, but needs nb (n = na + nb).
nb	Number of bottom time series; if C is present, nb is not used.
sparse	Option to return sparse object (*default* is TRUE).

Value

A list of five elements:

S	(n x nb) cross-sectional (contemporaneous) summing matrix.
Ut	(na x n) zero constraints cross-sectional (contemporaneous) kernel matrix.
n	Number of variables na + nb.
na	Number of upper level variables.
nb	Number of bottom level variables.

Examples

```
# One level hierarchy (na = 1, nb = 2)
obj <- hts_tools(C = matrix(c(1, 1), 1), sparse = FALSE)
```

---

iterec — *Iterative heuristic cross-temporal forecast reconciliation*

Description

Iterative procedure which produces cross-temporally reconciled forecasts by alternating forecast reconciliation along one single dimension (either cross-sectional or temporal) at each iteration step. **Each iteration** consists in the first two steps of the heuristic KA procedure, so the forecasts are reconciled by alternating cross-sectional (contemporaneous) reconciliation, and reconciliation through temporal hierarchies in a cyclic fashion. The choice of the dimension along with the first reconciliation step in each iteration is performed is up to the user (param start_rec), and there is no particular reason why one should perform the temporal reconciliation first, and the cross-sectional reconciliation then. The iterative procedure allows the user to get non-negative reconciled forecasts.

Usage

```
iterec(basef, m, C, thf_comb, hts_comb, Ut, nb, res, W,
       Omega, mse = TRUE, corpcor = FALSE,type = "M",
       sol = "direct", nn = FALSE, maxit = 100, tol = 1e-5,
       start_rec = "thf", note = TRUE, plot = "mti",
       settings = osqpSettings(verbose = FALSE, eps_abs = 1e-5,
       eps_rel = 1e-5, polish_refine_iter = 100, polish = TRUE))
```

Arguments

basef	(n x h(k* + m)) matrix of base forecasts to be reconciled; n is the total number of variables, m is the highest frequency, k* is the sum of (p-1) factors of m, excluding m, and h is the forecast horizon. Each row identifies, a time series, and the forecasts are ordered as [lowest_freq' ... highest_freq']'.
m	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation).

C
: (na x nb) cross-sectional (contemporaneous) matrix mapping the bottom level series into the higher level ones.

thf_comb
: Type of the ((k* + m) x (k* + m)) covariance matrix to be used in the temporal reconciliation, see more in comb param of thfrec.

hts_comb
: Type of the (n x n) covariance matrix to be used in the cross-sectional reconciliation, see more in comb param of htsrec.

Ut
: Zero constraints cross-sectional (contemporaneous) kernel matrix ($\mathbf{U}'\mathbf{Y} = \mathbf{0}$) spanning the null space valid for the reconciled forecasts. It can be used instead of parameter C, but in this case nb (n = na + nb) is needed. If the hierarchy admits a structural representation, Ut has dimension (na x n).

nb
: Number of bottom time series; if C is present, nb is not used.

res
: (n x N(k* + m)) matrix containing the residuals at all the temporal frequencies ordered [lowest_freq' ... highest_freq']' (columns) for each variable (row), needed to estimate the covariance matrix when hts_comb = {"wls", "shr", "sam"} and/or hts_comb = {"wlsv", "wlsh", "acov", "strar1", "sar1", "har1", "shr", "sam"}. The row must be in the same order as basef.

W
: This option permits to directly enter the covariance matrix in the cross-sectional reconciliation, see more in W param of htsrec.

Omega
: This option permits to directly enter the covariance matrix in the reconciliation through temporal hierarchies, see more in Omega param of thfrec.

mse
: Logical value: TRUE (*default*) calculates the covariance matrix of the in-sample residuals (when necessary) according to the original **hts** and **thief** formulation: no mean correction, T as denominator.

corpcor
: Logical value: TRUE if **corpcor** (Schäfer et al., 2017) must be used to shrink the sample covariance matrix according to Schäfer and Strimmer (2005), otherwise the function uses the same implementation as package **hts**.

type
: Approach used to compute the reconciled forecasts: "M" for the projection approach with matrix M (*default*), or "S" for the structural approach with summing matrix S.

sol
: Solution technique for the reconciliation problem: either "direct" for the direct solution or "osqp" for the numerical solution (solving a linearly constrained quadratic program using solve_osqp).

nn
: Logical value: TRUE if non-negative reconciled forecasts are wished.

maxit
: Max number of iteration (100, *default*).

tol
: Convergence tolerance (1e-5, *default*).

start_rec
: Dimension along with the first reconciliation step in each iteration is performed: it start from temporal reconciliation with "thf" (*default*), from cross-sectional with "hts" and it does both reconciliation with "auto".

note
: If note = TRUE (*default*) the function writes some notes to the console, otherwise no note is produced (also no plot).

plot
: Some useful plots: "mti" (*default*) marginal trend inconsistencies, "pat" step by step inconsistency pattern for each iteration, "distf" distance forecasts iteration i and i-1, "all" all the plots.

settings — Settings for **osqp** (object osqpSettings). The default options are: verbose = FALSE, eps_abs = 1e-5, eps_rel = 1e-5, polish_refine_iter = 100 and polish = TRUE. For details, see the **osqp** documentation (Stellato et al., 2019).

Details

The above procedure can be seen as an extension of the well known iterative proportional fitting procedure (Deming and Stephan, 1940, Johnston and Pattie, 1993), also known as RAS method (Miller and Blair, 2009), to adjust the internal cell values of a two-dimensional matrix until they sum to some predetermined row and column totals. In that case the adjustment follows a proportional adjustment scheme, whereas in the cross-temporal reconciliation framework each adjustment step is made according to the penalty function associated to the single-dimension reconciliation procedure adopted.

Control status of iterative reconciliation:

- -2 Temporal/Cross-sectional reconciliation does not work.
- -1 Convergence not achieved (maximum iteration limit reached).
- 0 Convergence achieved.
- +1 Convergence achieved: incoherence has increased in the next iteration (at least one time).
- +2 Convergence achieved: incoherence has increased in the next two or more iteration (at least one time).
- +3 The forecasts are already reconciled.

Value

iterec returns a list with:

- recf — (n x h(k* + m)) matrix of reconciled forecasts.
- d_cs — Cross-sectional incoherence at each iteration.
- d_te — Temporal incoherence at each iteration.
- start_rec — Starting coherence dimension (thf or hts).
- tol — Tolerance.
- flag — Control code (see *details*).
- time — Elapsed time.
- dist — If start_rec = "auto", matrix of distances of the forecasts reconciled from the base.

References

Deming, E., Stephan, F.F. (1940), On a least squares adjustment of a sampled frequency table when the expected marginal totals are known, *The Annals of Mathematical Statistics*, 11, 4, 427–444.

Di Fonzo, T., Girolimetto, D. (2020), Cross-Temporal Forecast Reconciliation: Optimal Combination Method and Heuristic Alternatives, Department of Statistical Sciences, University of Padua, arXiv:2006.08570.

Johnston, R.J., Pattie, C.J. (1993), Entropy-Maximizing and the Iterative Proportional Fitting Procedure, *The Professional Geographer*, 45, 3, 317–322.

Kourentzes, N., Athanasopoulos, G. (2019), Cross-temporal coherent forecasts for Australian tourism, *Annals of Tourism Research*, 75, 393-409.

Miller, R.E., Blair, P.D. (2009), Input-output analysis: foundations and extensions, 2nd edition, New York, Cambridge University Press.

Schäfer, J.L., Opgen-Rhein, R., Zuber, V., Ahdesmaki, M., Duarte Silva, A.P., Strimmer, K. (2017), *Package 'corpcor'*, R package version 1.6.9 (April 1, 2017), https://CRAN.R-project.org/package=corpcor.

Schäfer, J.L., Strimmer, K. (2005), A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics, *Statistical Applications in Genetics and Molecular Biology*, 4, 1.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S. (2018). OSQP: An Operator Splitting Solver for Quadratic Programs, arXiv:1711.08013.

Stellato, B., Banjac, G., Goulart, P., Boyd, S., Anderson, E. (2019), OSQP: Quadratic Programming Solver using the 'OSQP' Library, R package version 0.6.0.3 (October 10, 2019), https://CRAN.R-project.org/package=osqp.

Examples

```
data(FoReco_data)
obj <- iterec(FoReco_data$base, note=FALSE,
  m = 12, C = FoReco_data$C, thf_comb = "acov",
  hts_comb = "shr", res = FoReco_data$res, start_rec = "thf")
```

---

octrec *Optimal combination cross-temporal forecast reconciliation*

Description

Optimal (in least squares sense) combination cross-temporal forecast reconciliation. The reconciled forecasts are calculated either through a projection approach (Byron, 1978), or the equivalent structural approach by Hyndman et al. (2011).

Usage

```
octrec(basef, m, C, comb, res, Ut, nb, W, Sstruc,
       mse = TRUE, corpcor = FALSE, type = "M", sol = "direct",
       nn = FALSE, keep = "list",
       settings = osqpSettings(verbose = FALSE, eps_abs = 1e-5,
       eps_rel = 1e-5, polish_refine_iter = 100, polish=TRUE))
```

Arguments

basef
: (n x h(k* + m)) matrix of base forecasts to be reconciled; n is the total number of variables, m is the highest frequency, k* is the sum of (p-1) factors of m, excluding m, and h is the forecast horizon. Each row identifies, a time series, and the forecasts are ordered as [lowest_freq' ... highest_freq']'.

m
: Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation).

C
: (na x nb) cross-sectional (contemporaneous) matrix mapping the bottom level series into the higher level ones.

comb
: Type of the reconciliation, it corrispond to a different covariance matrix (n(k* + m) x n(k* + m)), k* is the sum of (p-1) factors of m (exclude m as factors) and n is the number of variables:

- **ols** (Identity);
- **struc** (Cross-temporal summing matrix, use the Sstruc param to reduce computation time);
- **wlsh** (Hierarchy variances matrix);
- **wlsv** (Series variances matrix);
- **bdshr** (Shrunk cross-covariance matrix, cross-sectional framework);
- **bdsam** (Sample cross-covariance matrix, cross-sectional framework);
- **acov** (Series auto-covariance matrix);
- **Sshr** (Series shrunk cross-covariance matrix);
- **Ssam** (Series cross-covariance matrix);
- **shr** (Shrunk cross-covariance matrix);
- **sam** (Sample cross-covariance matrix);
- **w** use your personal matrix W in param W.

res
: (n x N(k* + m)) matrix containing the residuals at all the temporal frequencies ordered [lowest_freq' ... highest_freq']' (columns) for each variable (row), needed to estimate the covariance matrix when comb = {"sam", "wlsv", "wlsh", "acov", "Ssam", "Sshr", "Sshr1", "shr"}.

Ut
: Zero constraints cross-sectional (contemporaneous) kernel matrix ($\mathbf{U}'\mathbf{Y} = \mathbf{0}$) spanning the null space valid for the reconciled forecasts. It can be used instead of parameter C, but in this case nb (n = na + nb) is needed. If the hierarchy admits a structural representation, Ut has dimension (na x n).

nb
: Number of bottom time series; if C is present, nb is not used.

W
: This option permits to directly enter the covariance matrix:

1. W must be a p.d. (n(k* + m) x n(k* + m)) matrix;
2. if comb is different from "w", W is not used.

Sstruc
: Cross-temporal summing matrix (structural representation), $\check{\mathbf{S}}$; can be obtained through the function ctf_tools.

mse
: Logical value: TRUE (*default*) calculates the covariance matrix of the in-sample residuals (when necessary) according to the original **hts** and **thief** formulation: no mean correction, T as denominator.

corpcor Logical value: TRUE if **corpcor** (Schäfer et al., 2017) must be used to shrink the sample covariance matrix according to Schäfer and Strimmer (2005), otherwise the function uses the same implementation as package **hts**.

type Approach used to compute the reconciled forecasts: "M" for the projection approach with matrix M (*default*), or "S" for the structural approach with summing matrix S.

sol Solution technique for the reconciliation problem: either "direct" (*default*) for the direct solution or "osqp" for the numerical solution (solving a linearly constrained quadratic program using solve_osqp).

nn Logical value: TRUE if non-negative reconciled forecasts are wished.

keep Return a list object of the reconciled forecasts at all levels.

settings Settings for **osqp** (object osqpSettings). The default options are: verbose = FALSE, eps_abs = 1e-5, eps_rel = 1e-5, polish_refine_iter = 100 and polish = TRUE. For details, see the **osqp** documentation (Stellato et al., 2019).

Details

In case of non-negativity constraints, there are two ways:

1. sol = "direct" and nn = TRUE: the base forecasts will be reconciled at first without non-negativity constraints, then, if negative reconciled values are present, the "osqp" solver is used.
2. sol = "osqp" and nn = TRUE: the base forecasts will be reconciled through the "osqp" solver.

Value

If the parameter keep is equal to "recf", then the function returns only the (n x h(k* + m)) reconciled forecasts matrix, otherwise (keep="all") it returns a list that mainly depends on what type of representation (type) and methodology (sol) have been used:

recf (n x h(k* + m)) reconciled forecasts matrix.

Omega Covariance matrix used for reconciled forecasts (vec(Y') representation).

W Covariance matrix used for reconciled forecasts (vec(Y) representation).

nn_check Number of negative values (if zero, there are no values below zero).

rec_check Logical value: has the hierarchy been respected?

M (type="M" and type="direct")
Projection matrix (projection approach).

G (type="S" and type="direct")
Projection matrix (structural approach).

S (type="S" and type="direct")
Cross-temporal summing matrix, **QŠ** (vec(Y') representation).

info (type="osqp")
matrix with some useful indicators (columns) for each forecast horizon h (rows): run time (run_time) number of iteration, norm of primal residual (pri_res), status of osqp's solution (status) and polish's status (status_polish).

References

Byron, R.P. (1978), The estimation of large social accounts matrices, *Journal of the Royal Statistical Society A*, 141, 3, 359-367.

Di Fonzo, T., Girolimetto, D. (2020), Cross-Temporal Forecast Reconciliation: Optimal Combination Method and Heuristic Alternatives, Department of Statistical Sciences, University of Padua, arXiv:2006.08570.

Schäfer, J.L., Opgen-Rhein, R., Zuber, V., Ahdesmaki, M., Duarte Silva, A.P., Strimmer, K. (2017), *Package 'corpcor'*, R package version 1.6.9 (April 1, 2017), https://CRAN.R-project.org/package=corpcor.

Schäfer, J.L., Strimmer, K. (2005), A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics, *Statistical Applications in Genetics and Molecular Biology*, 4, 1.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S. (2018). OSQP: An Operator Splitting Solver for Quadratic Programs, arXiv:1711.08013.

Stellato, B., Banjac, G., Goulart, P., Boyd, S., Anderson, E. (2019), OSQP: Quadratic Programming Solver using the 'OSQP' Library, R package version 0.6.0.3 (October 10, 2019), https://CRAN.R-project.org/package=osqp.

Examples

```
data(FoReco_data)
obj <- octrec(FoReco_data$base, m = 12, C = FoReco_data$C,
              comb = "bdshr", res = FoReco_data$res)
```

---

score_index *Measuring forecasting accuracy*

Description

Function to calculate the accuracy indices of the reconciled point forecasts of a cross-temporal (not only, see examples) system (more in Average relative accuracy indices).

Usage

```
score_index(recf, base, test, m, nb, type = "mse", compact = TRUE)
```

Arguments

`recf`	list of q (forecast origins) reconciled forecasts' matrices (`[n x h(k* + m)]` in the cross-temporal, `[h x n]` in cross-sectional and vectors of length `[h(k* + m)]` in temporal framework).
`base`	list of q (forecast origins) base forecasts' matrices (`[n x h(k* + m)]` in the cross-temporal, `[n x h]` in cross-sectional and `[h(k* + m) x 1]` in temporal framework).

test	list of q (forecast origins) test observations' matrices ([n x h(k* + m)] in the cross-temporal, [n x h] in cross-sectional and [h(k* + m) x 1] in temporal framework).
m	highest frequency of the forecasted time series.
nb	number of bottom time series in the cross-sectional framework.
type	type of accuracy measure ("mse" Mean Square Error, "rmse" Root Mean Square Error or "mae" Mean Absolute Error)
compact	if TRUE return only the summary matrix.

Value

It returns a summary table called Avg_mat (if compact option is TRUE, *default*), otherwise it returns a list of four tables (more in Average relative accuracy indices).

References

Di Fonzo, T., Girolimetto, D. (2020), Cross-Temporal Forecast Reconciliation: Optimal Combination Method and Heuristic Alternatives, Department of Statistical Sciences, University of Padua, arXiv:2006.08570.

Examples

```
data(FoReco_data)

# Cross-temporal framework
oct_recf <- octrec(FoReco_data$base, m = 12, C = FoReco_data$C,
                   comb = "bdshr", res = FoReco_data$res)$recf
oct_score <- score_index(recf = oct_recf,
                         base = FoReco_data$base,
                         test = FoReco_data$test, m = 12, nb = 5)

# Cross-sectional framework
# monthly base forecasts
id <- which(simplify2array(strsplit(colnames(FoReco_data$base), split = "_"))[1, ] == "k1")
mbase <- t(FoReco_data$base[, id])
# monthly test set
mtest <- t(FoReco_data$test[, id])
# monthly residuals
id <- which(simplify2array(strsplit(colnames(FoReco_data$res), split = "_"))[1, ] == "k1")
mres <- t(FoReco_data$res[, id])
# monthly reconciled forecasts
mrecf <- htsrec(mbase, C = FoReco_data$C, comb = "shr", res = mres)$recf
# score
hts_score <- score_index(recf = mrecf, base = mbase, test = mtest, nb = 5)

# Temporal framework
data(FoReco_data)
# top ts base forecasts ([lowest_freq' ...  highest_freq']')
topbase <- FoReco_data$base[1, ]
```

```
# top ts residuals ([lowest_freq' ...  highest_freq']')
topres <- FoReco_data$res[1, ]
# top ts test ([lowest_freq' ...  highest_freq']')
toptest <- FoReco_data$test[1, ]
# top ts recf ([lowest_freq' ...  highest_freq']')
toprecf <- thfrec(topbase, m = 12, comb = "acov", res = topres)$recf
# score
thf_score <- score_index(recf = toprecf, base = topbase, test = toptest, m = 12)
```

---

shrink_estim *Shrinkage of the covariance matrix*

Description

Shrinkage of the covariance matrix according to Schäfer and Strimmer (2005).

Usage

```
shrink_estim(x, minT = T)
```

Arguments

x	residual matrix
minT	this param allows to calculate the covariance matrix according to the original hts formulation (TRUE) or according to the standard approach (FALSE).

Value

A list with two objects: the first ($scov) is the shrunk covariance matrix and the second ($lambda) is the shrinkage intensity coefficient.

Author(s)

This function is a modified version of the shrink_estim() hidden function of **hts**.

References

Schäfer, J.L., Strimmer, K. (2005), A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics, *Statistical Applications in Genetics and Molecular Biology*, 4, 1

Hyndman, R. J., Lee, A., Wang, E., and Wickramasuriya, S. (2020). hts: Hierarchical and Grouped Time Series, *R package version 6.0.1*, https://CRAN.R-project.org/package=hts.

tcsrec *Heuristic first-temporal-then-cross-sectional cross-temporal forecast reconciliation*

Description

The cross-temporal forecast reconciliation procedure by Kourentzes and Athanasopoulos (2019) can be viewed as an ensemble forecasting procedure which exploits the simple averaging of different forecasts. First, for each time series the forecasts at any temporal aggregation order are reconciled using temporal hierarchies (thfrec), then time-by-time cross-sectional reconciliation is performed (htsrec). The projection matrices obtained at this step are then averaged and used to cross-sectionally reconcile the forecasts obtained at step 1, by this way fulfilling both cross-sectional and temporal constraints.

Usage

```
tcsrec(basef, m, C, thf_comb, hts_comb, Ut, nb, res, W, Omega,
       mse = TRUE, corpcor = FALSE, avg="KA", nn= FALSE,
       settings = osqpSettings(verbose = FALSE, eps_abs = 1e-5,
       eps_rel = 1e-5, polish_refine_iter = 100, polish = TRUE))
```

Arguments

basef: (n x h(k* + m)) matrix of base forecasts to be reconciled; n is the total number of variables, m is the highest frequency, k* is the sum of (p-1) factors of m, excluding m, and h is the forecast horizon. Each row identifies, a time series, and the forecasts are ordered as [lowest_freq' ... highest_freq']'.

m: Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation).

C: (na x nb) cross-sectional (contemporaneous) matrix mapping the bottom level series into the higher level ones.

thf_comb: Type of the ((k* + m) x (k* + m)) covariance matrix to be used in the temporal reconciliation, see more in comb param of thfrec.

hts_comb: Type of the (n x n) covariance matrix to be used in the cross-sectional reconciliation, see more in comb param of htsrec.

Ut: Zero constraints cross-sectional (contemporaneous) kernel matrix $\mathbf{U}'\mathbf{Y} = \mathbf{0}$ spanning the null space valid for the reconciled forecasts. It can be used instead of parameter C, but needs nb (n = na + nb). If the hierarchy admits a structural representation, Ut has dimension (na x n).

nb: Number of bottom time series; if C is present, nb is not used.

res: (n x N(k* + m)) matrix containing the residuals at all the temporal frequencies ordered [lowest_freq' ... highest_freq']' (columns) for each variable (row), needed to estimate the covariance matrix when hts_comb = {"wls", "shr", "sam"} and/or hts_comb = {"wlsv", "wlsh", "acov", "strar1", "sar1", "har1", "shr", "sam"}. The row must be in the same order as basef.

W	This option permits to directly enter the covariance matrix in the cross-sectional reconciliation, see more in W param of htsrec.
Omega	This option permits to directly enter the covariance matrix in the reconciliation through temporal hierarchies, see more in Omega param of thfrec.
mse	Logical value: TRUE (*default*) calculates the covariance matrix of the in-sample residuals (when necessary) according to the original **hts** and **thief** formulation: no mean correction, T as denominator.
corpcor	Logical value: TRUE if **corpcor** (Schäfer et al., 2017) must be used to shrink the sample covariance matrix according to Schäfer and Strimmer (2005), otherwise the function uses the same implementation as package **hts**.
avg	If avg = "KA" (*default*), the final projection matrix M is the one proposed by Kourentzes and Athanasopoulos (2019), otherwise it is calculated as simple average of all the involved projection matrices at step 2 of th procedure (see Di Fonzo and Girolimetto, 2020).
nn	Logical value, TRUE if non-negative reconciled forecasts are wished. **Warning**, the two-step heuristic reconciliation allows non negativity constraints only in the first step. This means that non-negativity is not guaranteed in the final reconciled values.
settings	Settings for **osqp** (object osqpSettings). The default options are: verbose = FALSE, eps_abs = 1e-5, eps_rel = 1e-5, polish_refine_iter = 100 and polish = TRUE. For details, see the **osqp** documentation (Stellato et al., 2019).

Details

This function performs a two-step cross-temporal forecast reconciliation using the covariance matrices chosen by the user. If the combinations used by Kourentzes and Athanasopoulos (2019) are wished, thf_comb must be set equal to either "struc" or "wlsv", and hts_comb equal to either "shr" or "wls".

Value

The function returns a list with two elements:

recf	(n x h(k* + m)) reconciled forecasts matrix.
M	Projection matrix (projection approach).

References

Di Fonzo, T., Girolimetto, D. (2020), Cross-Temporal Forecast Reconciliation: Optimal Combination Method and Heuristic Alternatives, Department of Statistical Sciences, University of Padua, arXiv:2006.08570.

Kourentzes, N., Athanasopoulos, G. (2019), Cross-temporal coherent forecasts for Australian tourism, *Annals of Tourism Research*, 75, 393-409.

Schäfer, J.L., Opgen-Rhein, R., Zuber, V., Ahdesmaki, M., Duarte Silva, A.P., Strimmer, K. (2017), *Package 'corpcor'*, R package version 1.6.9 (April 1, 2017), https://CRAN.R-project.org/package=corpcor.

Schäfer, J.L., Strimmer, K. (2005), A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics, *Statistical Applications in Genetics and Molecular Biology*, 4, 1.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S. (2018). OSQP: An Operator Splitting Solver for Quadratic Programs, arXiv:1711.08013.

Stellato, B., Banjac, G., Goulart, P., Boyd, S., Anderson, E. (2019), OSQP: Quadratic Programming Solver using the 'OSQP' Library, R package version 0.6.0.3 (October 10, 2019), https://CRAN.R-project.org/package=osqp.

Examples

```
data(FoReco_data)
obj <- tcsrec(FoReco_data$base, m = 12, C = FoReco_data$C, thf_comb = "acov",
              hts_comb = "shr", res = FoReco_data$res)
```

---

thfrec *Forecast reconciliation through temporal hierarchies (temporal reconciliation)*

Description

Forecast reconciliation of one time series through temporal hierarchies (Athanasopoulos et al., 2017). The reconciled forecasts are calculated either through a projection approach (Byron, 1978), or the equivalent structural approach by Hyndman et al. (2011). Moreover, the classic bottom-up approach is available.

Usage

```
thfrec(basef, m, comb, res, mse = TRUE, corpcor = FALSE, Omega,
       type = "M", sol = "direct", nn = FALSE, keep = "list",
       settings = osqpSettings(verbose = FALSE, eps_abs = 1e-5,
       eps_rel = 1e-5, polish_refine_iter = 100, polish=TRUE))
```

Arguments

`basef` Vector of base forecasts to be reconciled, containing the forecasts at all the needed temporal frequencies ordered as [lowest_freq' ... highest_freq']'.

`m` Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation).

`comb` Type of the reconciliation. Except for bottom up, all other options correspond to a different `((k* + m) x (k* + m))` covariance matrix, `k*` is the sum of `(p-1)` factors of `m` (excluding `m`):

- **bu** (Bottom-up);
- **ols** (Identity);
- **struc** (Structural variances);

- **wlsv** (Series variances);
- **wlsh** (Hierarchy variances);
- **acov** (Auto-covariance matrix);
- **strar1** (Structural Markov);
- **sar1** (Series Markov);
- **har1** (Hierarchy Markov);
- **shr** (Shrunk cross-covariance matrix);
- **sam** (Sample cross-covariance matrix);
- **omega** use your personal matrix Omega in param `Omega`.

`res` vector containing the in-sample residuals at all the temporal frequencies ordered as `basef`, i.e. [lowest_freq' ... highest_freq']', needed to estimate the covariance matrix when `comb = {"wlsv", "wlsh", "acov", "strar1", "sar1", "har1", "shr", "sam"}`.

`mse` Logical value: `TRUE` (*default*) calculates the covariance matrix of the in-sample residuals (when necessary) according to the original **hts** and **thief** formulation: no mean correction, T as denominator.

`corpcor` Logical value: `TRUE` if **corpcor** (Schäfer et al., 2017) must be used to shrink the sample covariance matrix according to Schäfer and Strimmer (2005), otherwise the function uses the same implementation as package **hts**.

`Omega` This option permits to directly enter the covariance matrix:

1. `Omega` must be a p.d. `((k* + m) x (k* + m))` matrix;
2. if `comb` is different from `"omega"`, `Omega` is not used.

`type` Approach used to compute the reconciled forecasts: `"M"` for the projection approach with matrix M (*default*), or `"S"` for the structural approach with summing matrix S.

`sol` Solution technique for the reconciliation problem: either `"direct"` (*default*) for the direct solution or `"osqp"` for the numerical solution (solving a linearly constrained quadratic program using `solve_osqp`).

`nn` Logical value: `TRUE` if non-negative reconciled forecasts are wished.

`keep` Return a list object of the reconciled forecasts at all levels.

`settings` Settings for **osqp** (object `osqpSettings`). The default options are: `verbose = FALSE`, `eps_abs = 1e-5`, `eps_rel = 1e-5`, `polish_refine_iter = 100` and `polish = TRUE`. For details, see the **osqp** documentation (Stellato et al., 2019).

Details

In case of non-negativity constraints, there are two ways:

1. `sol = "direct"` and `nn = TRUE`: the base forecasts will be reconciled at first without non-negativity constraints, then, if negative reconciled values are present, the `"osqp"` solver is used.
2. `sol = "osqp"` and `nn = TRUE`: the base forecasts will be reconciled through the `"osqp"` solver.

Value

If the parameter keep is equal to "recf", then the function returns only the reconciled forecasts vector, otherwise (keep="all") it returns a list that mainly depends on what type of representation (type) and methodology (sol) have been used:

recf — (h(k* + m) x 1) reconciled forecasts vector.

Omega — Covariance matrix used for forecast reconciliation.

nn_check — Number of negative values (if zero, there are no values below zero).

rec_check — Logical value: has the hierarchy been respected?

M (type="M" and type="direct") — Projection matrix (projection approach)

G (type="S" and type="direct") — Projection matrix (structural approach).

S (type="S" and type="direct") — Temporal summing matrix, **R**.

info (type="osqp") — matrix with some useful indicators (columns) for each forecast horizon h (rows): run time (run_time) number of iteration, norm of primal residual (pri_res), status of osqp's solution (status) and polish's status (status_polish).

Only if comb = "bu", the function returns recf, S and M.

References

Athanasopoulos, G., Hyndman, R.J., Kourentzes, N., Petropoulos, F. (2017), Forecasting with Temporal Hierarchies, *European Journal of Operational Research*, 262, 1, 60-74.

Byron, R.P. (1978), The estimation of large social accounts matrices, *Journal of the Royal Statistical Society A*, 141, 3, 359-367.

Di Fonzo, T., Girolimetto, D. (2020), Cross-Temporal Forecast Reconciliation: Optimal Combination Method and Heuristic Alternatives, Department of Statistical Sciences, University of Padua, arXiv:2006.08570.

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G., Shang, H.L. (2011), Optimal combination forecasts for hierarchical time series, *Computational Statistics & Data Analysis*, 55, 9, 2579-2589.

Nystrup, P., Lindström, E., Pinson, P., Madsen, H. (2020), Temporal hierarchies with autocorrelation for load forecasting, *European Journal of Operational Research*, 280, 1, 876-888.

Schäfer, J.L., Opgen-Rhein, R., Zuber, V., Ahdesmaki, M., Duarte Silva, A.P., Strimmer, K. (2017), Package 'corpcor', R package version 1.6.9 (April 1, 2017), https://CRAN.R-project.org/package=corpcor.

Schäfer, J.L., Strimmer, K. (2005), A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics, *Statistical Applications in Genetics and Molecular Biology*, 4, 1.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S. (2018). OSQP: An Operator Splitting Solver for Quadratic Programs, arXiv:1711.08013.

Stellato, B., Banjac, G., Goulart, P., Boyd, S., Anderson, E. (2019), OSQP: Quadratic Programming Solver using the 'OSQP' Library, R package version 0.6.0.3 (October 10, 2019), https://CRAN.R-project.org/package=osqp.

Examples

```
data(FoReco_data)
# top ts base forecasts ([lowest_freq' ...  highest_freq']')
topbase <- FoReco_data$base[1, ]
 # top ts residuals ([lowest_freq' ...  highest_freq']')
topres <- FoReco_data$res[1, ]
obj <- thfrec(topbase, m = 12, comb = "acov", res = topres)
```

---

thf_tools   *Temporal reconciliation tools*

Description

Some useful tools for forecast reconciliation through temporal hierarchies.

Usage

```
thf_tools(m, h = 1, sparse = TRUE)
```

Arguments

m	Highest available sampling frequency per seasonal cycle (max. order of temporal aggregation).
h	Forecast horizon for the lowest frequency (most temporally aggregated) time series (*default* is 1).
sparse	Option to return sparse object (*default* is TRUE).

Value

A list of seven elements:

K	Temporal aggregation matrix.
R	Temporal summing matrix.
Zt	Zero constraints temporal kernel matrix, $\mathbf{Z}_h'\mathbf{Y}' = \mathbf{0}_{[hk^* \times n]}$.
kset	Set of factors (p) of m in descending order (from m to 1), $\mathcal{K} = \{k_p, k_{p-1}, \ldots, k_2, k_1\}$, $k_p = m,\ k_1 = 1$.
p	Number of elements of kset, $(\mathcal{K})$.
ks	Sum of p-1 factors of m (out of m itself), k^*.
kt	Sum of all factors of m ($k^{tot} = k^* + m$).

Examples

```
# quarterly data
obj <- thf_tools(m = 4, sparse = FALSE)
```

Index