Subset Selection for Forecast Reconciliation

Xiaoqian Wang

2023-01-17

1 Hierarchical time series

Create a hierarchical structure as complete as possible. Only nested structure is considered here. Then the summing matrix $\boldsymbol{S}$ is given and $\hat{\boldsymbol{y}}_h$ can be obtained easily by specifying forecasting models.

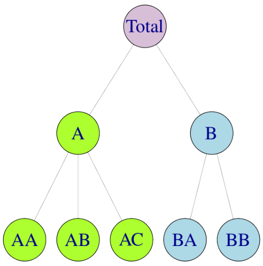

Figure 1: A 2-level hierarchical tree structure

For example,

$$\boldsymbol{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } \hat{\boldsymbol{y}}_h = (10, 6, 5, 1, 4, 0, 2, 5)'.$$

2 Linear forecast reconciliation

$$\tilde{\boldsymbol{y}}_h = \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_h,$$

where $\boldsymbol{G}_h$ maps the base forecasts $\hat{\boldsymbol{y}}_h$ into the bottom level, it combines all base forecasts to form bottom-level forecasts.

- **Bottom-up approach:** $\boldsymbol{G}_{bu} = \begin{bmatrix} \boldsymbol{O}_{n_b \times n_a} & \boldsymbol{I}_{n_b} \end{bmatrix}$
- **Top-down approach:** $\boldsymbol{G}_{td} = \begin{bmatrix} \boldsymbol{p} & \boldsymbol{O}_{n_b \times (n-1)} \end{bmatrix}$, where $\boldsymbol{p}$ is an n_b-dimensional vector including

the set of disaggregation proportions.

- **Middle-out approach:** $\boldsymbol{G}_{mo} = \begin{bmatrix} \boldsymbol{O}_{n_b \times n_t} & \boldsymbol{P}_{n_b \times n_l} & \boldsymbol{O}_{n_b \times n_d} \end{bmatrix}$, where $n_t + n_l + n_d = n$, $n_d \geq n_b$, $n_t \geq 1$, $\sum_{i=1}^{n_b} \sum_{j=1}^{n_l} p_{ij} = n_l$.
- **Optimization approaches:** $\boldsymbol{G}_h = \left(\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S}\right)^{-1} \boldsymbol{S}'\boldsymbol{W}_h^{-1}$ is obtained by solving the optimization problem

$$\min_{\boldsymbol{G}_h} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_h\right)' \boldsymbol{W}_h^{-1} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_h\right), \text{ s.t. } \boldsymbol{G}_h\boldsymbol{S} = \boldsymbol{I}_{n_b}$$

3 Subset selection

a. About $\boldsymbol{G}_h$. Our goal is to **zero out some columns** so that the corresponding base forecasts in $\hat{\boldsymbol{y}}_h$ are not used to form the reconciled bottom-level forecasts and, furthermore, all reconciled forecasts.

b. About $\boldsymbol{S}$. It sums up the reconciled bottom-level forecasts to get the full set of reconciled forecasts. For the purpose of automatic selection and comparison, we don't need to zero out the corresponding rows.

We have to add additional constraints on $\boldsymbol{G}_h$. This leads to a main question:

How to mathematically describe the constraint?

3.1 Solution 1

$$\tilde{\boldsymbol{y}}_h = \boldsymbol{S}\boldsymbol{G}_h\boldsymbol{B}_h\hat{\boldsymbol{y}}_h,$$

where $\boldsymbol{B}_h$ is an $n \times n$ diagonal matrix with elements of the main diagonal being either zero or one.

The optimization problem can be written as

$$\begin{aligned} \min\nolimits_{\boldsymbol{G}_h} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\boldsymbol{G}_h\boldsymbol{B}_h\hat{\boldsymbol{y}}_h\right)' \boldsymbol{W}_h^{-1} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\boldsymbol{G}_h\boldsymbol{B}_h\hat{\boldsymbol{y}}_h\right) + \lambda ||\boldsymbol{B}_h\boldsymbol{1}||_1, \\ \text{s.t. } \boldsymbol{G}_h\boldsymbol{B}_h\boldsymbol{S} = \boldsymbol{I}_{n_b} \\ b_{ii}(1-b_{ii}) = 0 \text{ for } i = 1, 2, \ldots, n \\ b_{ij} = 0 \text{ for } i \neq j \end{aligned}$$

The difficulty is that we wish to find the value of $\boldsymbol{G}_h$ and $\boldsymbol{B}_h$ simultaneously.

3.2 Solution 2

$$\tilde{\boldsymbol{y}}_h = \boldsymbol{S}\check{\boldsymbol{G}}_h\check{\boldsymbol{B}}_h\hat{\boldsymbol{y}}_h,$$

where $\check{\boldsymbol{G}}_h$ is given using ols, wls, structural scaling, or shrinkage estimator of $\boldsymbol{W}_h$. $\check{\boldsymbol{B}}_h$ is an $n \times n$ diagonal matrix with elements of the main diagonal being either greater than or equal to zero. So $\check{\boldsymbol{B}}_h$

can adaptively changes given $\check{\boldsymbol{G}}_h$ and we can only estimate $\check{\boldsymbol{B}}_h$. But the reconciled forecasts are likely to no longer preserve the unbiasedness.

The optimization problem can be written as

$$\begin{aligned}\min\nolimits_{\boldsymbol{G}_h} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\check{\boldsymbol{G}}_h\check{\boldsymbol{B}}_h\hat{\boldsymbol{y}}_h\right)' \boldsymbol{W}_h^{-1} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\check{\boldsymbol{G}}_h\check{\boldsymbol{B}}_h\hat{\boldsymbol{y}}_h\right) + \lambda||\check{\boldsymbol{B}}_h\mathbf{1}||_1,\\ \text{s.t. } b_{ii} \geq 0 \text{ for } i = 1, \ldots, n\\ b_{ij} = 0 \text{ for } i \neq j\end{aligned}$$

3.2.1 R implementation

```r
library(CVXR)

y_hat <- c(10, 6, 5, 1, 4, 0, 2, 5)
S <- rbind(matrix(c(rep(1, 5), c(rep(1, 3), rep(0, 2)), c(rep(0, 3), rep(1, 2))),
                  nrow = 3, byrow = TRUE),
           diag(1, nrow = 5))

W_inv <- solve(diag(c(5,3,2,1,1,1,1,1))) # structural scaling approximation
G <- solve(t(S) %*% W_inv %*% S) %*% t(S) %*% W_inv

b <- Variable(8)
lambda <- 0.3

e <- y_hat - S %*% G  %*% diag(b) %*% y_hat
obj <- quad_form(e, W_inv) + lambda * norm1(b)
prob <- Problem(Minimize(obj), constraints = list(b >= 0))
res <- solve(prob)
res$getValue(b)
```

```
##               [,1]
## [1,]  2.076187e+00
## [2,]  6.941964e-05
## [3,]  9.090869e-01
## [4,]  6.774912e-01
## [5,]  9.474978e-01
## [6,]  4.014566e-12
## [7,] -4.811095e-05
## [8,]  5.999817e-01
```

```
(B_check <- diag(round(as.vector(res$getValue(b)), digits = 3)))
```

```
##       [,1] [,2]  [,3]  [,4]  [,5] [,6] [,7] [,8]
## [1,] 2.076    0 0.000 0.000 0.000    0    0  0.0
## [2,] 0.000    0 0.000 0.000 0.000    0    0  0.0
## [3,] 0.000    0 0.909 0.000 0.000    0    0  0.0
## [4,] 0.000    0 0.000 0.677 0.000    0    0  0.0
## [5,] 0.000    0 0.000 0.000 0.947    0    0  0.0
## [6,] 0.000    0 0.000 0.000 0.000    0    0  0.0
## [7,] 0.000    0 0.000 0.000 0.000    0    0  0.0
## [8,] 0.000    0 0.000 0.000 0.000    0    0  0.6
```

```
(y_tilde <- as.vector(S %*% G %*% B_check %*% y_hat))
```

```
## [1] 10.923333  5.183500  5.739833  0.916500  4.027500  0.239500  1.369917
## [8]  4.369917
```

3.2.2 Other issues

- $\boldsymbol{G}_h$ is the optimal solution of the minimization problem, while $\check{\boldsymbol{G}}_h\check{\boldsymbol{B}}_h$ is the suboptimal solution since we get $\check{\boldsymbol{B}}_h$ given $\check{\boldsymbol{G}}_h$.
- If we do subset selection, it is important to take into account the forecasting performance of each time series. Thus, maybe we can replace the first term with the errors of the reconciled forecasts using expanding/rolling windows.

3.3 Solution 3

$$\tilde{\boldsymbol{y}}_h = \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_h$$

The optimization problem can be written as

$$\begin{aligned} \min\nolimits_{\boldsymbol{G}_h} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_h\right)' \boldsymbol{W}_h^{-1} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_h\right), \\ \text{s.t. } \boldsymbol{G}_h\boldsymbol{S} = \boldsymbol{I}_{n_b} \\ ||\boldsymbol{G}_{\cdot j}||_1 < c \text{ for a sufficiently-small value } c > 0, j \in \{1, 2, \ldots n\} \end{aligned}$$

It is hard to define the constraint to shrink some columns of $\boldsymbol{G}$ to zero: (1) the number of zero-columns, and (2) If we add a penalty term $\lambda\sum_{j=1}^{n}||g_{\cdot j}||_1$, it is the same as $\lambda||\operatorname{vec}\boldsymbol{G}||_1$ which aims to shrink the elements of $\boldsymbol{G}$ to zero (Ben Taieb & Koo, 2019).

According to Miyashiro & Takano (2015), we can rewrite the problem as

$$\begin{aligned}\min_{\boldsymbol{G}_h} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_h\right)' \boldsymbol{W}_h^{-1} \left(\hat{\boldsymbol{y}}_h - \boldsymbol{S}\boldsymbol{G}_h\hat{\boldsymbol{y}}_h\right) + 2\left(\textstyle\sum_{j=1}^{n} b_j + 1\right) - n, \\ \text{s.t. } \boldsymbol{G}_h\boldsymbol{S} = \boldsymbol{I}_{n_b} \\ b_j = 0 \Rightarrow ||\boldsymbol{G}_{\cdot j}||_1 = 0 \quad (j = 1, 2, \ldots, n) \\ b_j \in \{0, 1\} \quad (j = 1, 2, \ldots, n)\end{aligned}$$