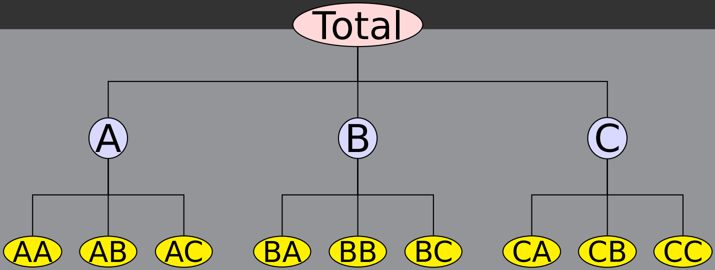
Rob J Hyndman

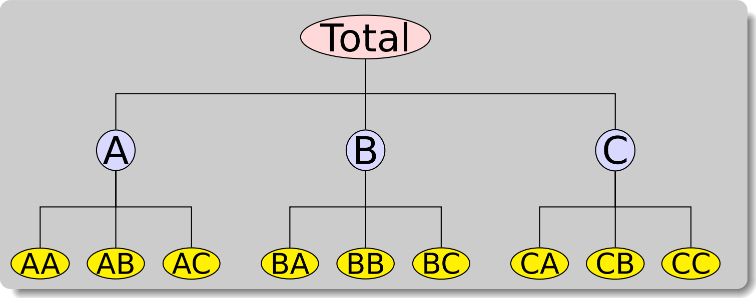
Forecasting hierarchical time series



Outline

1. **Hierarchical and grouped time series**
2. Optimal forecasts
3. Approximately optimal forecasts
4. Fast computation
5. Example: Australian tourism
6. Example: Temporal hierarchies
7. hts package for R
8. References

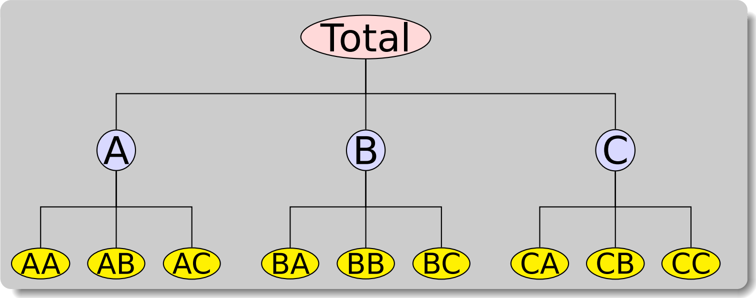
Introduction



Examples

- Manufacturing product hierarchies
- Net labour turnover
- Pharmaceutical sales
- Tourism demand by region and purpose

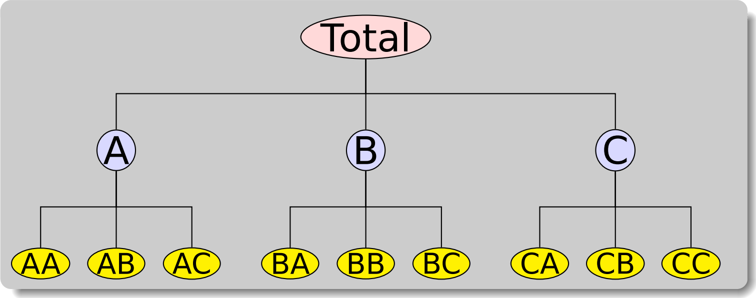
Introduction



Examples

- Manufacturing product hierarchies
- Net labour turnover
- Pharmaceutical sales
- Tourism demand by region and purpose

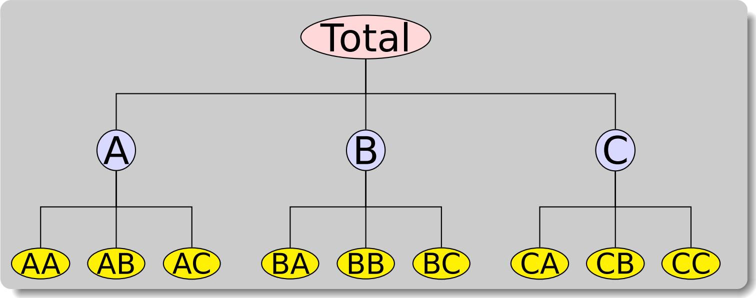
Introduction



Examples

- Manufacturing product hierarchies
- Net labour turnover
- Pharmaceutical sales
- Tourism demand by region and purpose

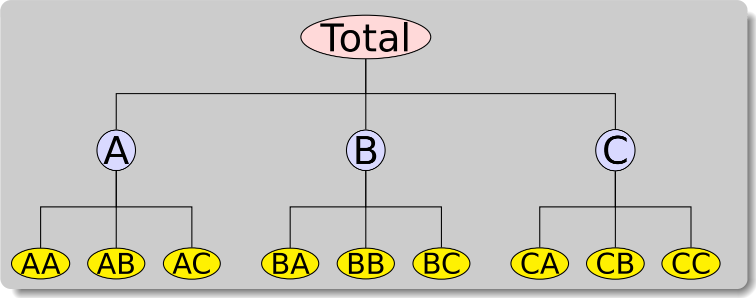
Introduction



Examples

- Manufacturing product hierarchies
- Net labour turnover
- Pharmaceutical sales
- Tourism demand by region and purpose

Introduction



Examples

- Manufacturing product hierarchies
- Net labour turnover
- Pharmaceutical sales
- Tourism demand by region and purpose

Forecasting the PBS

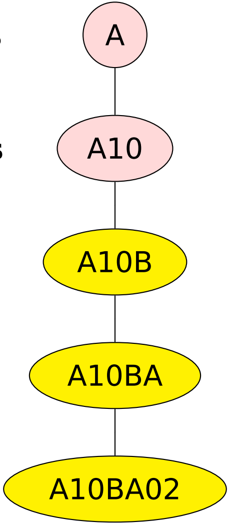
ATC drug classification

A	Alimentary tract and metabolism
B	Blood and blood forming organs
C	Cardiovascular system
D	Dermatologicals
G	Genito-urinary system and sex hormones
H	Systemic hormonal preparations, excluding sex hormones and insulins
J	Anti-infectives for systemic use
L	Antineoplastic and immunomodulating agents
M	Musculo-skeletal system
N	Nervous system
P	Antiparasitic products, insecticides and repellents
R	Respiratory system
S	Sensory organs
V	Various

ATC drug classification

14 classes

84 classes



Alimentary tract and metabolism

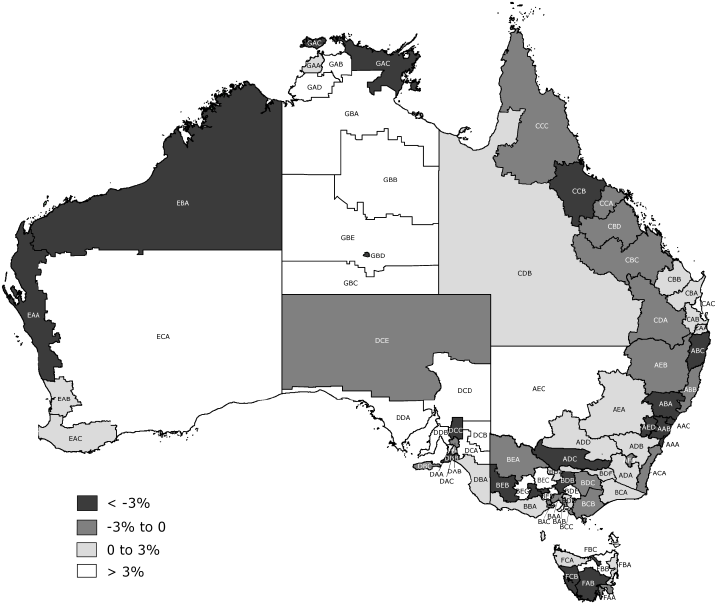
Drugs used in diabetes

Blood glucose lowering drugs

Biguanides

Metformin

Australian tourism



Australian tourism

Also split by purpose of travel:

- Holiday
- Visits to friends and relatives
- Business
- Other

Hierarchical/grouped time series

- A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

Example: Pharmaceutical products are organized in a hierarchy under the Anatomical Therapeutic Chemical (ATC) Classification System.

- A **grouped time series** is a collection of time series that are aggregated in a number of non-hierarchical ways.

Example: Australian tourism demand is grouped by region and purpose of travel.

Hierarchical/grouped time series

- A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

Example: Pharmaceutical products are organized in a hierarchy under the Anatomical Therapeutic Chemical (ATC) Classification System.

- A **grouped time series** is a collection of time series that are aggregated in a number of non-hierarchical ways.

Example: Australian tourism demand is grouped by region and purpose of travel.

Hierarchical/grouped time series

- A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

Example: Pharmaceutical products are organized in a hierarchy under the Anatomical Therapeutic Chemical (ATC) Classification System.

- A **grouped time series** is a collection of time series that are aggregated in a number of non-hierarchical ways.

Example: Australian tourism demand is grouped by region and purpose of travel.

Hierarchical/grouped time series

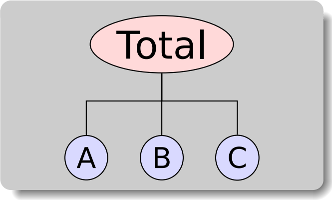
- A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

Example: Pharmaceutical products are organized in a hierarchy under the Anatomical Therapeutic Chemical (ATC) Classification System.

- A **grouped time series** is a collection of time series that are aggregated in a number of non-hierarchical ways.

Example: Australian tourism demand is grouped by region and purpose of travel.

Hierarchical data

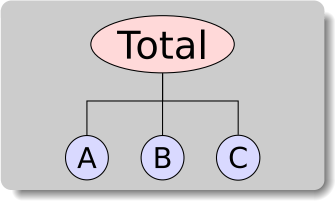


Y_t : observed aggregate of all series at time t.

$Y_{X,t}$: observation on series X at time t.

$\boldsymbol{B}_t$: vector of all series at bottom level in time t.

Hierarchical data

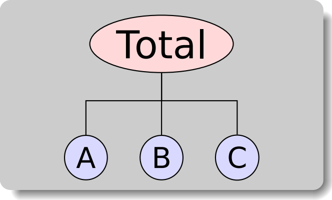


Y_t : observed aggregate of all series at time t.

$Y_{X,t}$: observation on series X at time t.

$\boldsymbol{B}_t$: vector of all series at bottom level in time t.

Hierarchical data



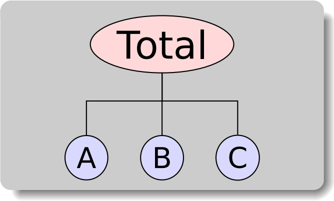
Y_t : observed aggregate of all series at time t.

$Y_{X,t}$: observation on series X at time t.

$\boldsymbol{B}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{Y}_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix}$$

Hierarchical data



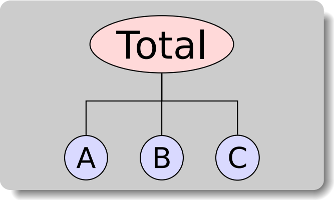
Y_t : observed aggregate of all series at time t.

$Y_{X,t}$: observation on series X at time t.

$\boldsymbol{B}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{Y}_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix}$$

Hierarchical data



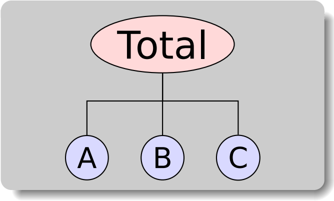
Y_t : observed aggregate of all series at time t.

$Y_{X,t}$: observation on series X at time t.

$\boldsymbol{B}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{Y}_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix}}_{\boldsymbol{B}_t}$$

Hierarchical data



Y_t : observed aggregate of all series at time t.

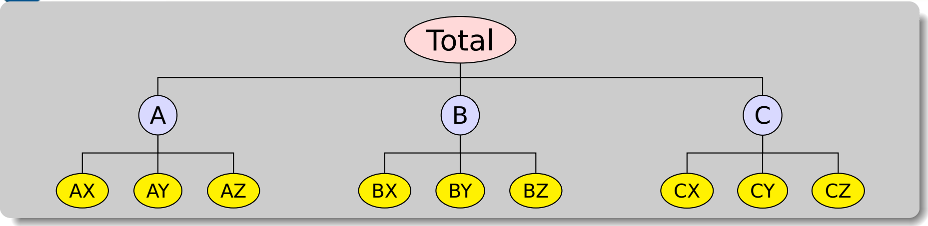
$Y_{X,t}$: observation on series X at time t.

$\boldsymbol{B}_t$: vector of all series at bottom level in time t.

$$\boldsymbol{Y}_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]' = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix}}_{\boldsymbol{B}_t}$$

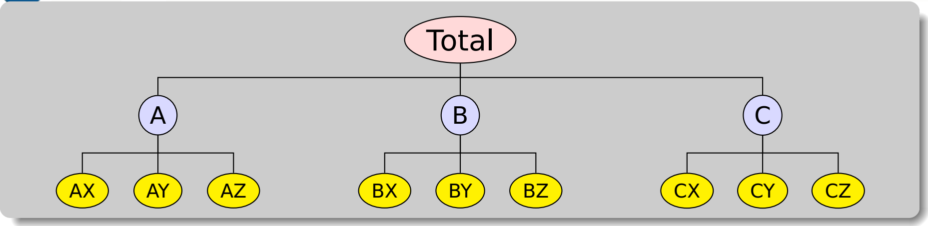
$$\boxed{\boldsymbol{Y}_t = \boldsymbol{S}\boldsymbol{B}_t}$$

Hierarchical data



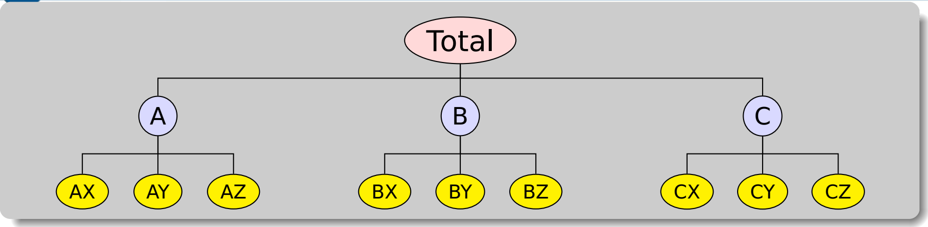
$$\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix}}_{\boldsymbol{B}_t}$$

Hierarchical data



$$
\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix}}_{\boldsymbol{B}_t}
$$

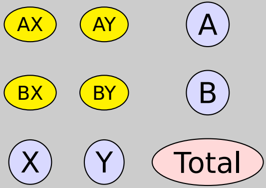
Hierarchical data



$$\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix}}_{\boldsymbol{B}_t}$$

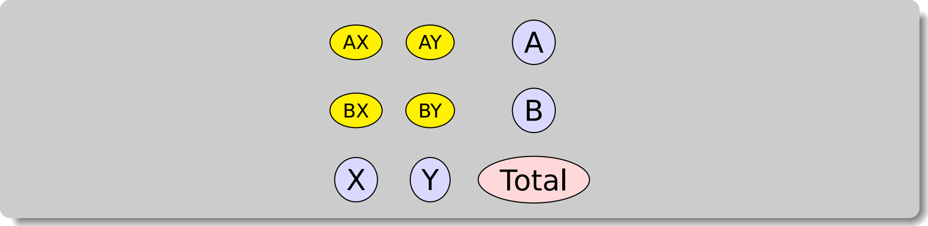
$$\boxed{\boldsymbol{Y}_t = \boldsymbol{S}\boldsymbol{B}_t}$$

Grouped data



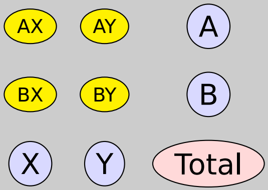
$$\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{X,t} \\ Y_{Y,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix}}_{\boldsymbol{B}_t}$$

Grouped data



$$\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{X,t} \\ Y_{Y,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix}}_{\boldsymbol{B}_t}$$

Grouped data



$$
\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{X,t} \\ Y_{Y,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix}}_{\boldsymbol{B}_t}
$$

$$
\boldsymbol{Y}_t = \boldsymbol{S}\boldsymbol{B}_t
$$

Outline

1 Hierarchical and grouped time series

2 **Optimal forecasts**

3 Approximately optimal forecasts

4 Fast computation

5 Example: Australian tourism

6 Example: Temporal hierarchies

7 hts package for R

8 References

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{B}_t$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = \boldsymbol{S}\boldsymbol{B}_t$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

Forecasts

Key idea: forecast reconciliation

- ➥ Ignore structural constraints and forecast every series of interest independently.
- ➥ Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{B}_t$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = \mathrm{E}[\boldsymbol{B}_{n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.

Forecasts

Key idea: forecast reconciliation

- ➥ Ignore structural constraints and forecast every series of interest independently.
- ➥ Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boxed{\boldsymbol{Y}_t = S\boldsymbol{B}_t}$. So $\boxed{\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h}$.

- $\beta_n(h) = \mathrm{E}[\boldsymbol{B}_{n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- ε_h has zero mean and covariance Σ_h.
- Estimate $\beta_n(h)$ using GLS?

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{B}_t$. So $\hat{\boldsymbol{Y}}_n(h) = S\boldsymbol{\beta}_n(h) + \boldsymbol{\varepsilon}_h$.

- $\boldsymbol{\beta}_n(h) = \mathrm{E}[\boldsymbol{B}_{n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- $\boldsymbol{\varepsilon}_h$ has zero mean and covariance Σ_h.
- Estimate $\boldsymbol{\beta}_n(h)$ using GLS?

Forecasts

Key idea: forecast reconciliation

- ➥ Ignore structural constraints and forecast every series of interest independently.
- ➥ Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{B}_t$. So $\hat{\boldsymbol{Y}}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = \mathrm{E}[\boldsymbol{B}_{n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- ε_h has zero mean and covariance Σ_h.
- Estimate $\beta_n(h)$ using GLS?

Forecasts

Key idea: forecast reconciliation

- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

Let $\hat{\boldsymbol{Y}}_n(h)$ be vector of initial h-step forecasts, made at time n, stacked in same order as $\boldsymbol{Y}_t$.

$\boldsymbol{Y}_t = S\boldsymbol{B}_t$. So $\hat{\boldsymbol{Y}}_n(h) = S\boldsymbol{\beta}_n(h) + \boldsymbol{\varepsilon}_h$.

- $\boldsymbol{\beta}_n(h) = \mathrm{E}[\boldsymbol{B}_{n+h} \mid \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n]$.
- $\boldsymbol{\varepsilon}_h$ has zero mean and covariance Σ_h.
- Estimate $\boldsymbol{\beta}_n(h)$ using GLS?

Optimal combination forecasts

$$\boxed{\tilde{\mathbf{Y}}_n(h) = \mathbf{S}\hat{\beta}_n(h) = \mathbf{S}(\mathbf{S}'\Sigma_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'\Sigma_h^{\dagger}\hat{\mathbf{Y}}_n(h)}$$

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}\hat{\beta}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

Initial forecasts

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = \mathbf{S}\hat{\beta}_n(h) = \mathbf{S}(\mathbf{S}'\Sigma_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'\Sigma_h^{\dagger}\hat{\mathbf{Y}}_n(h)$$

Revised forecasts

Initial forecasts

- $\Sigma_h^{\dagger}$ is generalized inverse of Σ_h.
- $\text{Var}[\tilde{\mathbf{Y}}_n(h)|\mathbf{Y}_1,\dots,\mathbf{Y}_n] = \mathbf{S}(\mathbf{S}'\Sigma_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'$
- **Problem:** Σ_h hard to estimate.

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}\hat{\beta}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

Revised forecasts

Initial forecasts

- $\Sigma_h^{\dagger}$ is generalized inverse of Σ_h.
- $\text{Var}[\tilde{\boldsymbol{Y}}_n(h)|\boldsymbol{Y}_1,\dots,\boldsymbol{Y}_n] = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'$
- **Problem:** Σ_h hard to estimate.

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}\hat{\beta}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

Revised forecasts

Initial forecasts

- $\Sigma_h^{\dagger}$ is generalized inverse of Σ_h.
- $\mathrm{Var}[\tilde{\boldsymbol{Y}}_n(h)|\boldsymbol{Y}_1,\dots,\boldsymbol{Y}_n] = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'$
- **Problem:** Σ_h hard to estimate.

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}\hat{\beta}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^\dagger \boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^\dagger \hat{\boldsymbol{Y}}_n(h)$$

Revised forecasts

Initial forecasts

- $\Sigma_h^\dagger$ is generalized inverse of Σ_h.
- $\text{Var}[\tilde{\boldsymbol{Y}}_n(h)|\boldsymbol{Y}_1,\dots,\boldsymbol{Y}_n] = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^\dagger \boldsymbol{S})^{-1}\boldsymbol{S}'$
- **Problem:** Σ_h hard to estimate.

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = \mathbf{S}\hat{\beta}_n(h) = \mathbf{S}(\mathbf{S}'\Sigma_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'\Sigma_h^{\dagger}\hat{\mathbf{Y}}_n(h)$$

Revised forecasts

Initial forecasts

- $\Sigma_h^{\dagger}$ is generalized inverse of Σ_h.
- $\text{Var}[\tilde{\mathbf{Y}}_n(h)|\mathbf{Y}_1,\dots,\mathbf{Y}_n] = \mathbf{S}(\mathbf{S}'\Sigma_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'$
- **Problem:** Σ_h hard to estimate.

Outline

1 Hierarchical and grouped time series

2 Optimal forecasts

3 **Approximately optimal forecasts**

4 Fast computation

5 Example: Australian tourism

6 Example: Temporal hierarchies

7 hts package for R

8 References

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

Revised forecasts

Base forecasts

Solution 1: OLS

- Assume $\varepsilon_h \approx \boldsymbol{S}\varepsilon_{B,h}$ where $\varepsilon_{B,h}$ is the forecast error at bottom level.
- Then $\Sigma_h \approx \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h = \text{Var}(\varepsilon_{B,h})$.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\Sigma_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{\dagger} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^\dagger \boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^\dagger \hat{\mathbf{Y}}_n(h)$$

Revised forecasts

Base forecasts

Solution 1: OLS

- Assume $\varepsilon_h \approx \boldsymbol{S}\varepsilon_{B,h}$ where $\varepsilon_{B,h}$ is the forecast error at bottom level.
- Then $\Sigma_h \approx \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h = \text{Var}(\varepsilon_{B,h})$.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\Sigma_h^\dagger\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^\dagger = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

$$\tilde{\mathbf{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^\dagger\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^\dagger\hat{\mathbf{Y}}_n(h)$$

Revised forecasts | Base forecasts

Solution 1: OLS

- Assume $\varepsilon_h \approx \boldsymbol{S}\varepsilon_{B,h}$ where $\varepsilon_{B,h}$ is the forecast error at bottom level.
- Then $\Sigma_h \approx \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h = \text{Var}(\varepsilon_{B,h})$.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\Sigma_h^\dagger\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^\dagger = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

$$\tilde{\mathbf{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = \mathbf{S}(\mathbf{S}'\Sigma_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'\Sigma_h^{\dagger}\hat{\mathbf{Y}}_n(h)$$

Revised forecasts Base forecasts

Solution 1: OLS

- Assume $\varepsilon_h \approx \mathbf{S}\varepsilon_{B,h}$ where $\varepsilon_{B,h}$ is the forecast error at bottom level.
- Then $\Sigma_h \approx \mathbf{S}\Omega_h\mathbf{S}'$ where $\Omega_h = \mathrm{Var}(\varepsilon_{B,h})$.
- If Moore-Penrose generalized inverse used, then $(\mathbf{S}'\Sigma_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'\Sigma_h^{\dagger} = (\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'$.

$$\tilde{\mathbf{Y}}_n(h) = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{\dagger}\hat{\mathbf{Y}}_n(h)$$

Revised forecasts

Base forecasts

Solution 1: OLS

- Assume $\varepsilon_h \approx \boldsymbol{S}\varepsilon_{B,h}$ where $\varepsilon_{B,h}$ is the forecast error at bottom level.
- Then $\Sigma_h \approx \boldsymbol{S}\Omega_h\boldsymbol{S}'$ where $\Omega_h = \text{Var}(\varepsilon_{B,h})$.
- If Moore-Penrose generalized inverse used, then $(\boldsymbol{S}'\Sigma_h^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_h^{\dagger} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.

$$\tilde{\mathbf{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\mathbf{Y}}_n(h) = \mathbf{S}(\mathbf{S}'\Sigma_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'\Sigma_h^{\dagger}\hat{\mathbf{Y}}_n(h)$$

Revised forecasts

Base forecasts

Solution 1: OLS

- Assume $\varepsilon_h \approx \mathbf{S}\varepsilon_{B,h}$ where $\varepsilon_{B,h}$ is the forecast error at bottom level.
- Then $\Sigma_h \approx \mathbf{S}\Omega_h\mathbf{S}'$ where $\Omega_h = \text{Var}(\varepsilon_{B,h})$.
- If Moore-Penrose generalized inverse used, then $(\mathbf{S}'\Sigma_h^{\dagger}\mathbf{S})^{-1}\mathbf{S}'\Sigma_h^{\dagger} = (\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'$.

$$\tilde{\mathbf{Y}}_n(h) = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\hat{\mathbf{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

- GLS = OLS.
- Optimal weighted average of initial forecasts.
- Optimal reconciliation weights are $\boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.
- Weights are independent of the data and of the covariance structure of the hierarchy!

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

- GLS = OLS.
- Optimal weighted average of initial forecasts.
- Optimal reconciliation weights are $\boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.
- Weights are independent of the data and of the covariance structure of the hierarchy!

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

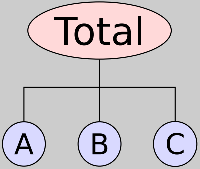
- GLS = OLS.
- Optimal weighted average of initial forecasts.
- Optimal reconciliation weights are $\boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.
- Weights are independent of the data and of the covariance structure of the hierarchy!

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

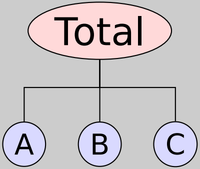
- GLS = OLS.
- Optimal weighted average of initial forecasts.
- Optimal reconciliation weights are $\boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'$.
- Weights are independent of the data and of the covariance structure of the hierarchy!

Optimal combination forecasts



$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

Optimal combination forecasts

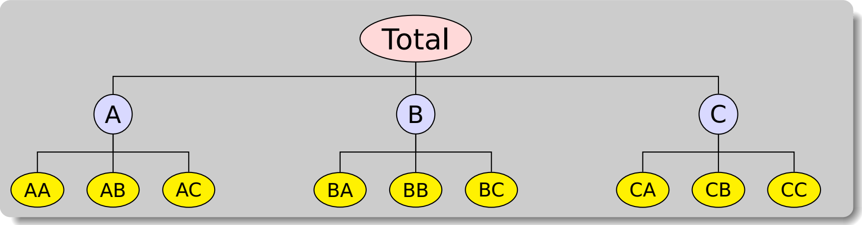


$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

Weights:

$$\boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}' = \begin{bmatrix} 0.75 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.75 & -0.25 & -0.25 \\ 0.25 & -0.25 & 0.75 & -0.25 \\ 0.25 & -0.25 & -0.25 & 0.75 \end{bmatrix}$$

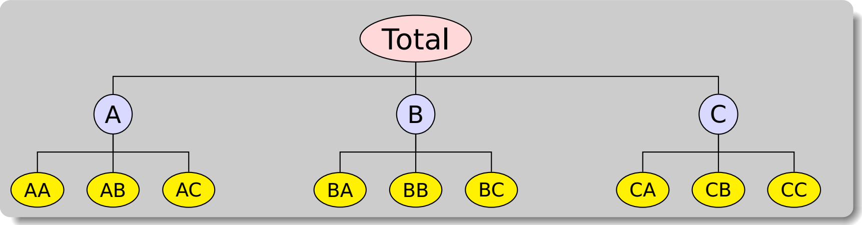
Optimal combination forecasts



Weights: $\boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}' =$

$$\begin{bmatrix}
0.69 & 0.23 & 0.23 & 0.23 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 \\
0.23 & 0.58 & -0.17 & -0.17 & 0.19 & 0.19 & 0.19 & -0.06 & -0.06 & -0.06 & -0.06 & -0.06 & -0.06 \\
0.23 & -0.17 & 0.58 & -0.17 & -0.06 & -0.06 & -0.06 & 0.19 & 0.19 & 0.19 & -0.06 & -0.06 & -0.06 \\
0.23 & -0.17 & -0.17 & 0.58 & -0.06 & -0.06 & -0.06 & -0.06 & -0.06 & -0.06 & 0.19 & 0.19 & 0.19 \\
0.08 & 0.19 & -0.06 & -0.06 & 0.73 & -0.27 & -0.27 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
0.08 & 0.19 & -0.06 & -0.06 & -0.27 & 0.73 & -0.27 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
0.08 & 0.19 & -0.06 & -0.06 & -0.27 & -0.27 & 0.73 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & 0.73 & -0.27 & -0.27 & -0.02 & -0.02 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & -0.27 & 0.73 & -0.27 & -0.02 & -0.02 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & -0.27 & -0.27 & 0.73 & -0.02 & -0.02 & -0.02 \\
0.08 & -0.06 & -0.06 & 0.19 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & 0.73 & -0.27 & -0.27 \\
0.08 & -0.06 & -0.06 & 0.19 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.27 & 0.73 & -0.27 \\
0.08 & -0.06 & -0.06 & 0.19 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.27 & -0.27 & 0.73
\end{bmatrix}$$

Optimal combination forecasts



Weights: $\boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}' =$

$$\begin{bmatrix}
0.69 & 0.23 & 0.23 & 0.23 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 \\
0.23 & 0.58 & -0.17 & -0.17 & 0.19 & 0.19 & 0.19 & -0.06 & -0.06 & -0.06 & -0.06 & -0.06 & -0.06 \\
0.23 & -0.17 & 0.58 & -0.17 & -0.06 & -0.06 & -0.06 & 0.19 & 0.19 & 0.19 & -0.06 & -0.06 & -0.06 \\
0.23 & -0.17 & -0.17 & 0.58 & -0.06 & -0.06 & -0.06 & -0.06 & -0.06 & -0.06 & 0.19 & 0.19 & 0.19 \\
0.08 & 0.19 & -0.06 & -0.06 & 0.73 & -0.27 & -0.27 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
0.08 & 0.19 & -0.06 & -0.06 & -0.27 & 0.73 & -0.27 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
0.08 & 0.19 & -0.06 & -0.06 & -0.27 & -0.27 & 0.73 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & 0.73 & -0.27 & -0.27 & -0.02 & -0.02 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & -0.27 & 0.73 & -0.27 & -0.02 & -0.02 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & -0.27 & -0.27 & 0.73 & -0.02 & -0.02 & -0.02 \\
0.08 & -0.06 & -0.06 & 0.19 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & 0.73 & -0.27 & -0.27 \\
0.08 & -0.06 & -0.06 & 0.19 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.27 & 0.73 & -0.27 \\
0.08 & -0.06 & -0.06 & 0.19 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.27 & -0.27 & 0.73
\end{bmatrix}$$

Features

- Covariates can be included in initial forecasts.
- Adjustments can be made to initial forecasts at any level.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Features

- Covariates can be included in initial forecasts.
- Adjustments can be made to initial forecasts at any level.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Features

- Covariates can be included in initial forecasts.
- Adjustments can be made to initial forecasts at any level.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Features

- Covariates can be included in initial forecasts.
- Adjustments can be made to initial forecasts at any level.
- Very simple and flexible method. Can work with *any* hierarchical or grouped time series.
- Conceptually easy to implement: OLS on base forecasts.

Challenges

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

- Computational difficulties in big hierarchies due to size of the $\boldsymbol{S}$ matrix and singular behavior of $(\boldsymbol{S}'\boldsymbol{S})$.
- Need to estimate covariance matrix to produce prediction intervals.
- Assumption might be unrealistic.
- Ignores covariance matrix in computing point forecasts.

Challenges

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

- Computational difficulties in big hierarchies due to size of the $\boldsymbol{S}$ matrix and singular behavior of $(\boldsymbol{S}'\boldsymbol{S})$.
- Need to estimate covariance matrix to produce prediction intervals.
- Assumption might be unrealistic.
- Ignores covariance matrix in computing point forecasts.

Challenges

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

- Computational difficulties in big hierarchies due to size of the $\boldsymbol{S}$ matrix and singular behavior of $(\boldsymbol{S}'\boldsymbol{S})$.
- Need to estimate covariance matrix to produce prediction intervals.
- Assumption might be unrealistic.
- Ignores covariance matrix in computing point forecasts.

Challenges

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}'\hat{\boldsymbol{Y}}_n(h)$$

- Computational difficulties in big hierarchies due to size of the $\boldsymbol{S}$ matrix and singular behavior of $(\boldsymbol{S}'\boldsymbol{S})$.
- Need to estimate covariance matrix to produce prediction intervals.
- Assumption might be unrealistic.
- Ignores covariance matrix in computing point forecasts.

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

Solution 1: OLS

- Approximate $\Sigma_1^{\dagger}$ by $c\mathbf{I}$.

Solution 2: Rescaling

- Suppose we approximate Σ_1 by its diagonal.
- Let $\Lambda = \left[\text{diagonal}(\Sigma_1)\right]^{-1}$ contain inverse one-step forecast variances.

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda\hat{\boldsymbol{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

Solution 1: OLS

- Approximate $\Sigma_1^{\dagger}$ by $c\mathbf{I}$.

Solution 2: Rescaling

- Suppose we approximate Σ_1 by its diagonal.
- Let $\Lambda = \left[\text{diagonal}\left(\Sigma_1\right)\right]^{-1}$ contain inverse one-step forecast variances.

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda\hat{\boldsymbol{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

Solution 1: OLS

- Approximate $\Sigma_1^{\dagger}$ by $c\mathbf{I}$.

Solution 2: Rescaling

- Suppose we approximate Σ_1 by its diagonal.
- Let $\boldsymbol{\Lambda} = \left[\text{diagonal}\left(\Sigma_1\right)\right]^{-1}$ contain inverse one-step forecast variances.

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda\hat{\boldsymbol{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

Solution 1: OLS

- Approximate $\boldsymbol{\Sigma}_1^{\dagger}$ by $c\mathbf{I}$.

Solution 2: Rescaling

- Suppose we approximate $\boldsymbol{\Sigma}_1$ by its diagonal.
- Let $\boldsymbol{\Lambda} = \left[\text{diagonal}\left(\boldsymbol{\Sigma}_1\right)\right]^{-1}$ contain inverse one-step forecast variances.

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda\hat{\boldsymbol{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Sigma_1^{\dagger}\boldsymbol{S})^{-1}\boldsymbol{S}'\Sigma_1^{\dagger}\hat{\boldsymbol{Y}}_n(h)$$

Solution 1: OLS

- Approximate $\Sigma_1^{\dagger}$ by $c\mathbf{I}$.

Solution 2: Rescaling

- Suppose we approximate Σ_1 by its diagonal.
- Let $\Lambda = \left[\text{diagonal}\left(\Sigma_1\right)\right]^{-1}$ contain inverse one-step forecast variances.

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda\hat{\boldsymbol{Y}}_n(h)$$

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda\hat{\boldsymbol{Y}}_n(h)$$

Solution 3: Averaging

- If the bottom level error series are approximately uncorrelated and have similar variances, then Λ is inversely proportional to the number of series contributing to each node.
- So set Λ to be the inverse row sums of $\boldsymbol{S}$.

Optimal combination forecasts

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda\hat{\boldsymbol{Y}}_n(h)$$

Solution 3: Averaging

- If the bottom level error series are approximately uncorrelated and have similar variances, then Λ is inversely proportional to the number of series contributing to each node.
- So set Λ to be the inverse row sums of $\boldsymbol{S}$.

Outline

1 Hierarchical and grouped time series

2 Optimal forecasts

3 Approximately optimal forecasts

4 **Fast computation**

5 Example: Australian tourism

6 Example: Temporal hierarchies

7 hts package for R

8 References

Fast computation

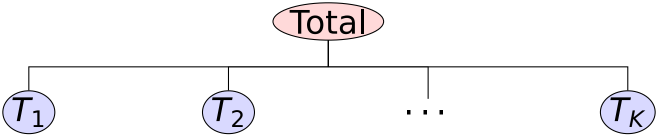
To efficiently compute the reconciled forecasts for large hierarchies or groups of time series, we must compute

$$\tilde{\boldsymbol{Y}}_n(h) = \boldsymbol{S}(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}\boldsymbol{S}'\Lambda\hat{\boldsymbol{Y}}_n(h)$$

without explicitly forming $\boldsymbol{S}$ or $(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1}$ or $\boldsymbol{S}'\Lambda$.

Fast computation: hierarchies

Think of the hierarchy as a tree of trees:



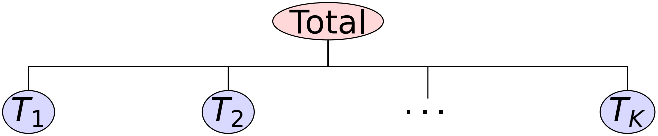
Then the summing matrix contains k smaller summing matrices:

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{1}'_{n_1} & \mathbf{1}'_{n_2} & \cdots & \mathbf{1}'_{n_K} \\ \boldsymbol{S}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{S}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{S}_K \end{bmatrix}$$

where $\mathbf{1}_n$ is an n-vector of ones and tree T_i has n_i terminal nodes.

Fast computation: hierarchies

Think of the hierarchy as a tree of trees:



Then the summing matrix contains k smaller summing matrices:

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{1}'_{n_1} & \boldsymbol{1}'_{n_2} & \cdots & \boldsymbol{1}'_{n_K} \\ \boldsymbol{S}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{S}_K \end{bmatrix}$$

where $\boldsymbol{1}_n$ is an n-vector of ones and tree T_i has n_i terminal nodes.

Fast computation: hierarchies

$$\boldsymbol{S}'\Lambda\boldsymbol{S} = \lambda_0\boldsymbol{J}_n + \begin{bmatrix} \boldsymbol{S}_1'\Lambda_1\boldsymbol{S}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}_2'\Lambda_2\boldsymbol{S}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{S}_K'\Lambda_K\boldsymbol{S}_K \end{bmatrix}$$

- λ_0 is the top left element of Λ;
- Λ_k is a block of Λ, corresponding to tree T_k;
- $\boldsymbol{J}_n$ is a matrix of ones;
- $n = \sum_k n_k$.

Now apply the Sherman-Morrison formula . . .

Fast computation: hierarchies

$$\boldsymbol{S}'\Lambda\boldsymbol{S} = \lambda_0\boldsymbol{J}_n + \begin{bmatrix} \boldsymbol{S}_1'\Lambda_1\boldsymbol{S}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{S}_2'\Lambda_2\boldsymbol{S}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{S}_K'\Lambda_K\boldsymbol{S}_K \end{bmatrix}$$

- λ_0 is the top left element of Λ;
- Λ_k is a block of Λ, corresponding to tree T_k;
- $\boldsymbol{J}_n$ is a matrix of ones;
- $n = \sum_k n_k$.

Now apply the Sherman-Morrison formula ...

Fast computation: hierarchies

$$(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1} = \begin{bmatrix} (\boldsymbol{S}_1'\Lambda_1\boldsymbol{S}_1)^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\boldsymbol{S}_2'\Lambda_2\boldsymbol{S}_2)^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (\boldsymbol{S}_K'\Lambda_K\boldsymbol{S}_K)^{-1} \end{bmatrix} - c\boldsymbol{S}_0$$

- $\boldsymbol{S}_0$ can be partitioned into K^2 blocks, with the (k,ℓ) block (of dimension $n_k \times n_\ell$) being

$$(\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k)^{-1}\boldsymbol{J}_{n_k,n_\ell}(\boldsymbol{S}_\ell'\Lambda_\ell\boldsymbol{S}_\ell)^{-1}$$

- $\boldsymbol{J}_{n_k,n_\ell}$ is a $n_k \times n_\ell$ matrix of ones.
- $c^{-1} = \lambda_0^{-1} + \sum_k \mathbf{1}_{n_k}'(\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k)^{-1}\mathbf{1}_{n_k}$.
- Each $\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k$ can be inverted similarly.
- $\boldsymbol{S}'\Lambda\boldsymbol{Y}$ can also be computed recursively.

Fast computation: hierarchies

$$(\boldsymbol{S}'\Lambda\boldsymbol{S})^{-1} = \begin{bmatrix} (\boldsymbol{S}_1'\Lambda_1\boldsymbol{S}_1)^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\boldsymbol{S}_2'\Lambda_2\boldsymbol{S}_2)^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (\boldsymbol{S}_K'\Lambda_K\boldsymbol{S}_K)^{-1} \end{bmatrix} - c\boldsymbol{S}_0$$

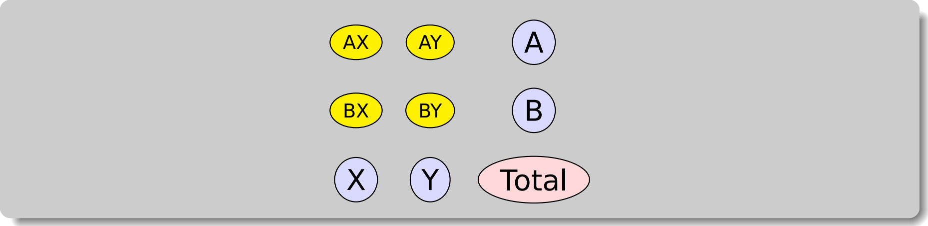
- $\boldsymbol{S}_0$ can be partitioned into K^2 blocks, with the (k,ℓ) block (of dimension $n_k \times n_\ell$) being

The recursive calculations can be done in such a way that we never store any of the large matrices involved.

- $\boldsymbol{J}_{n_k,n_\ell}$
- c^{-1}

- Each $\boldsymbol{S}_k'\Lambda_k\boldsymbol{S}_k$ can be inverted similarly.
- $\boldsymbol{S}'\Lambda\boldsymbol{Y}$ can also be computed recursively.

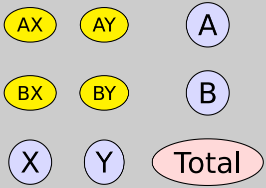
Fast computation: grouped data



$$\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{X,t} \\ Y_{Y,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix}}_{\boldsymbol{B}_t}$$

$$\boldsymbol{Y}_t = \boldsymbol{S}\boldsymbol{B}_t$$

Fast computation: grouped data



$$\boldsymbol{Y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{X,t} \\ Y_{Y,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix}}_{\boldsymbol{B}_t}$$

$$\boxed{\boldsymbol{Y}_t = \boldsymbol{S}\boldsymbol{B}_t}$$

Fast computation: grouped data

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{1}'_m \otimes \mathbf{1}'_n \\ \mathbf{1}'_m \otimes \mathbf{I}_n \\ \mathbf{I}_m \otimes \mathbf{1}'_n \\ \mathbf{I}_m \otimes \mathbf{I}_n \end{bmatrix}$$

$m =$ number of rows
$n =$ number of columns

$$\boldsymbol{S}'\Lambda\boldsymbol{S} = \lambda_{00}\boldsymbol{J}_{mn} + (\Lambda_R \otimes \boldsymbol{J}_n) + (\boldsymbol{J}_m \otimes \Lambda_C) + \Lambda_U$$

- Λ_R, Λ_C and Λ_U are diagonal matrices corresponding to rows, columns and unaggregated series;
- λ_{00} corresponds to aggregate.

Fast computation: grouped data

$$\boldsymbol{S} = \begin{bmatrix} \mathbf{1}'_m \otimes \mathbf{1}'_n \\ \mathbf{1}'_m \otimes \mathbf{I}_n \\ \mathbf{I}_m \otimes \mathbf{1}'_n \\ \mathbf{I}_m \otimes \mathbf{I}_n \end{bmatrix}$$

m = number of rows
n = number of columns

$$\boldsymbol{S}'\Lambda\boldsymbol{S} = \lambda_{00}\boldsymbol{J}_{mn} + (\Lambda_R \otimes \boldsymbol{J}_n) + (\boldsymbol{J}_m \otimes \Lambda_C) + \Lambda_U$$

- Λ_R, Λ_C and Λ_U are diagonal matrices corresponding to rows, columns and unaggregated series;
- λ_{00} corresponds to aggregate.

Fast computation: grouped data

$$(\boldsymbol{S}\Lambda\boldsymbol{S})^{-1} = \boldsymbol{A} - \frac{\boldsymbol{A}\boldsymbol{1}_{mn}\boldsymbol{1}_{mn}'\boldsymbol{A}}{1/\lambda_{00} + \boldsymbol{1}_{mn}'\boldsymbol{A}\boldsymbol{1}_{mn}}$$

$\boldsymbol{A} = \Lambda_U^{-1} - \Lambda_U^{-1}(\boldsymbol{J}_m \otimes \boldsymbol{D})\Lambda_U^{-1} - \boldsymbol{E}\boldsymbol{M}^{-1}\boldsymbol{E}'$.

$\boldsymbol{D}$ is diagonal with elements $d_j = \lambda_{0j}/(1 + \lambda_{0j}\sum_i \lambda_{ij}^{-1})$.

$\boldsymbol{E}$ has $m \times m$ blocks where $\boldsymbol{e}_{ij}$ has kth element

$$(\boldsymbol{e}_{ij})_k = \begin{cases} \lambda_{i0}^{1/2}\lambda_{ik}^{-1} - \lambda_{i0}^{1/2}\lambda_{ik}^{-2}d_k, & i = j, \\ -\lambda_{j0}^{1/2}\lambda_{ik}^{-1}\lambda_{jk}^{-1}d_k, & i \neq j. \end{cases}$$

$\boldsymbol{M}$ is $m \times m$ with (i,j) element

$$(\boldsymbol{M})_{ij} = \begin{cases} 1 + \lambda_{i0}\sum_k \lambda_{ik}^{-1} - \lambda_{i0}\sum_k \lambda_{ik}^{-2}d_k, & i = j, \\ -\lambda_{i0}^{1/2}\lambda_{j0}^{1/2}\sum_k \lambda_{ik}^{-1}\lambda_{jk}^{-1}d_k, & i \neq j. \end{cases}$$

Fast computation: grouped data

$$(\boldsymbol{S}\Lambda\boldsymbol{S})^{-1} = \boldsymbol{A} - \frac{\boldsymbol{A}\boldsymbol{1}_{mn}\boldsymbol{1}'_{mn}\boldsymbol{A}}{1/\lambda_{00} + \boldsymbol{1}'_{mn}\boldsymbol{A}\boldsymbol{1}_{mn}}$$

$\boldsymbol{A} = \Lambda_U^{-1} - \Lambda_U^{-1}(\boldsymbol{J}_m \otimes \boldsymbol{D})\Lambda_U^{-1} - \boldsymbol{E}\boldsymbol{M}^{-1}\boldsymbol{E}'$.

$\boldsymbol{D}$ is diagonal with elements $d_j = \lambda_{0j}/(1 + \lambda_{0j}\sum_i \lambda_{ij}^{-1})$.

$\boldsymbol{E}$ has $m \times m$ blocks where $\boldsymbol{e}_{ij}$ has kth element

Again, the calculations can be done in such a way that we never store any of the large matrices involved.

$\boldsymbol{M}$ is $m \times m$ with (i,j) element

$$(\boldsymbol{M})_{ij} = \begin{cases} 1 + \lambda_{i0}\sum_k \lambda_{ik}^{-1} - \lambda_{i0}\sum_k \lambda_{ik}^{-2} d_k, & i = j, \\ -\lambda_{i0}^{1/2}\lambda_{j0}^{1/2}\sum_k \lambda_{ik}^{-1}\lambda_{jk}^{-1} d_k, & i \neq j. \end{cases}$$

Fast computation

When the time series are not strictly hierarchical and have more than two grouping variables:

- Use sparse matrix storage and arithmetic.
- Use iterative approximation for inverting large sparse matrices.

(Paige and Saunders, 1982).

Fast computation

When the time series are not strictly hierarchical and have more than two grouping variables:

- Use sparse matrix storage and arithmetic.
- Use iterative approximation for inverting large sparse matrices.

(Paige and Saunders, 1982).

Fast computation

When the time series are not strictly hierarchical and have more than two grouping variables:

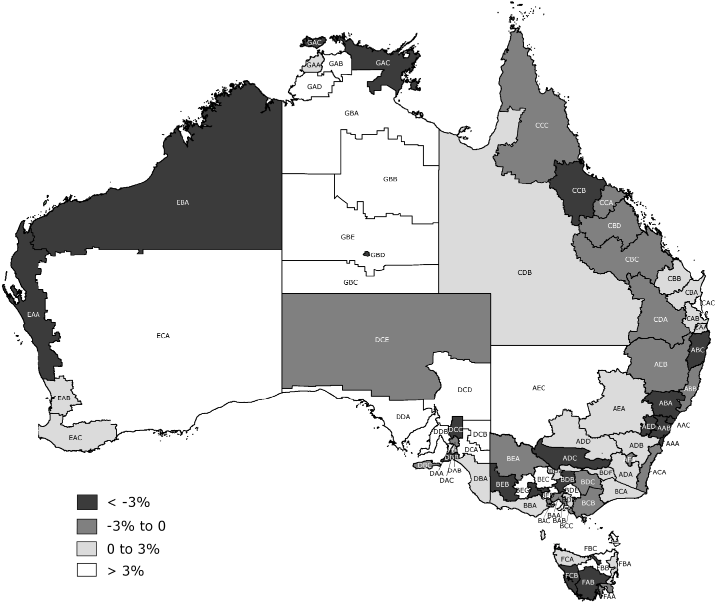
- Use sparse matrix storage and arithmetic.
- Use iterative approximation for inverting large sparse matrices.

(Paige and Saunders, 1982).

Outline

1 Hierarchical and grouped time series
2 Optimal forecasts
3 Approximately optimal forecasts
4 Fast computation
5 Example: Australian tourism
6 Example: Temporal hierarchies
7 hts package for R
8 References

Australian tourism



Australian tourism

Domestic visitor nights

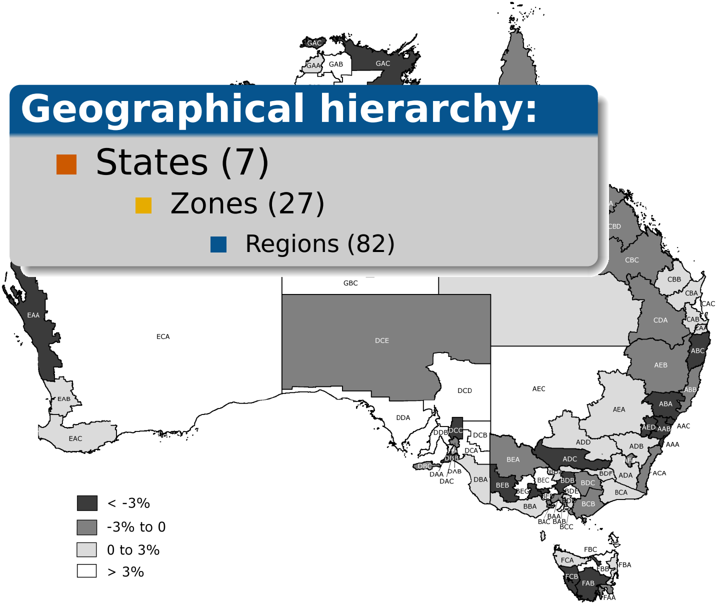
Quarterly data: 1998 – 2006.

From: *National Visitor Survey*, based on annual interviews of 120,000 Australians aged 15+, collected by Tourism Research Australia.

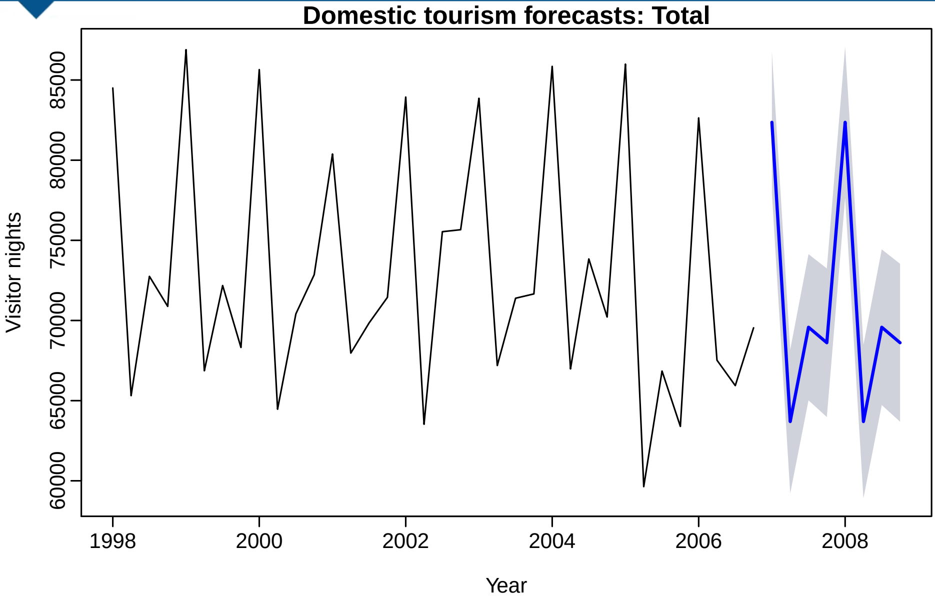
Australian tourism

Geographical hierarchy:

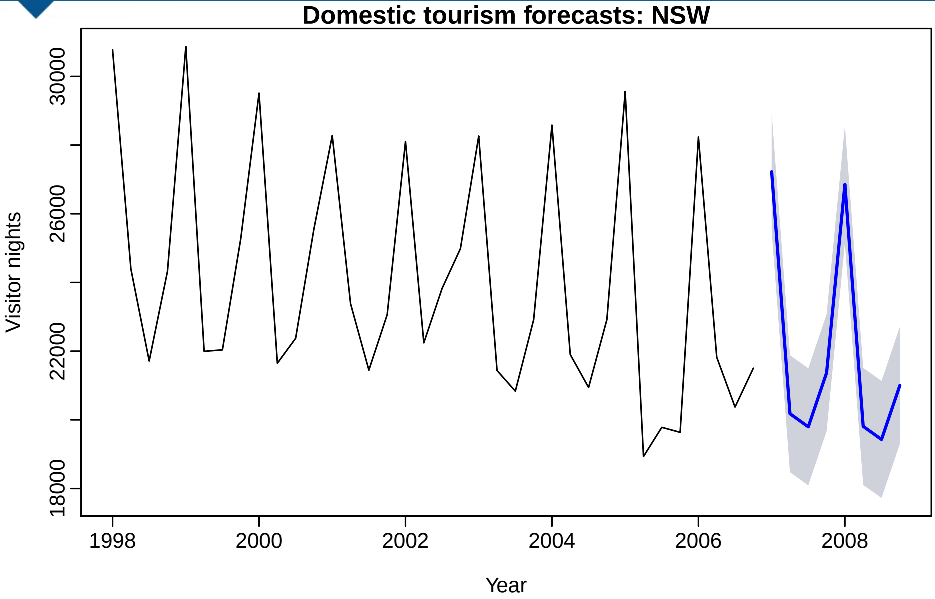
- States (7)
  - Zones (27)
    - Regions (82)



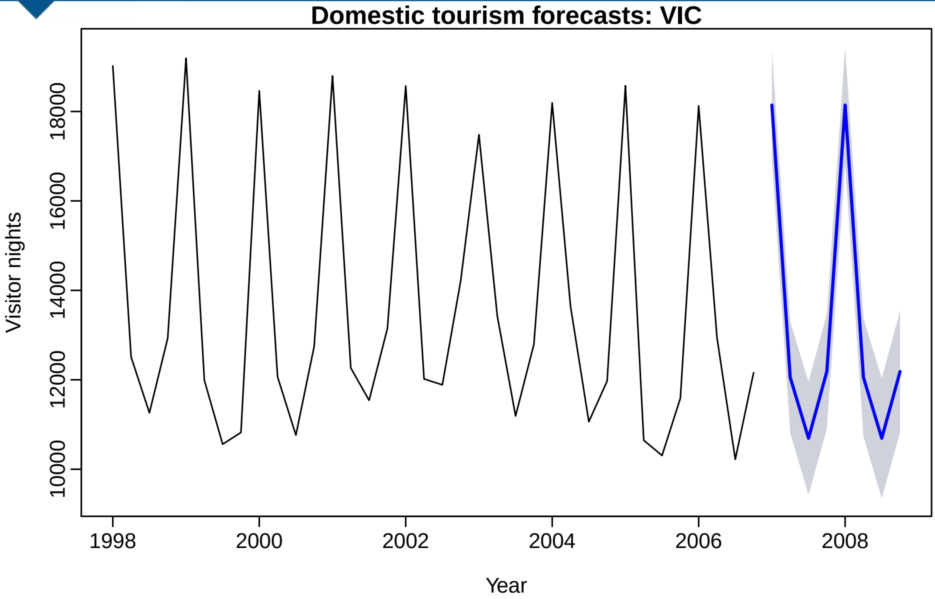
Base forecasts



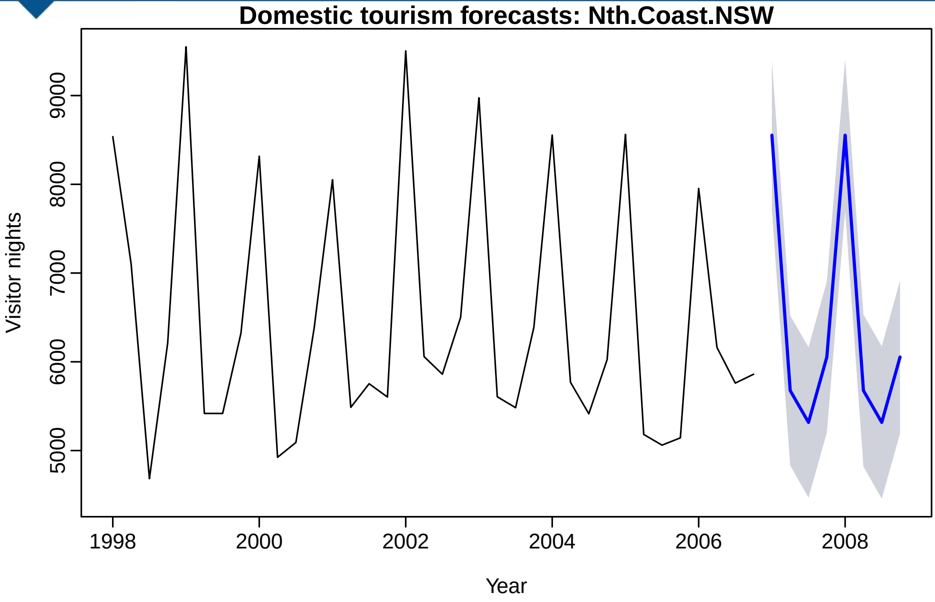
Base forecasts



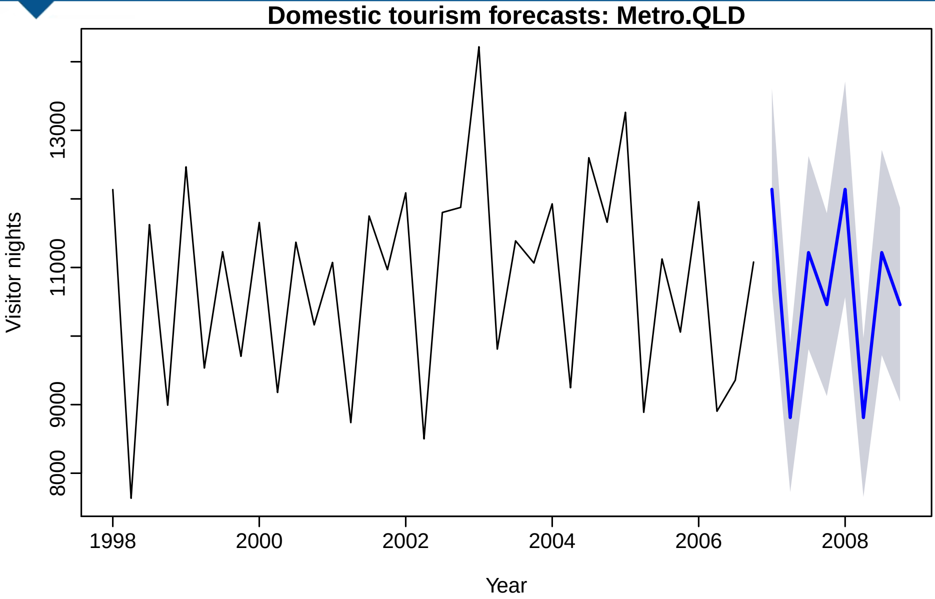
Base forecasts



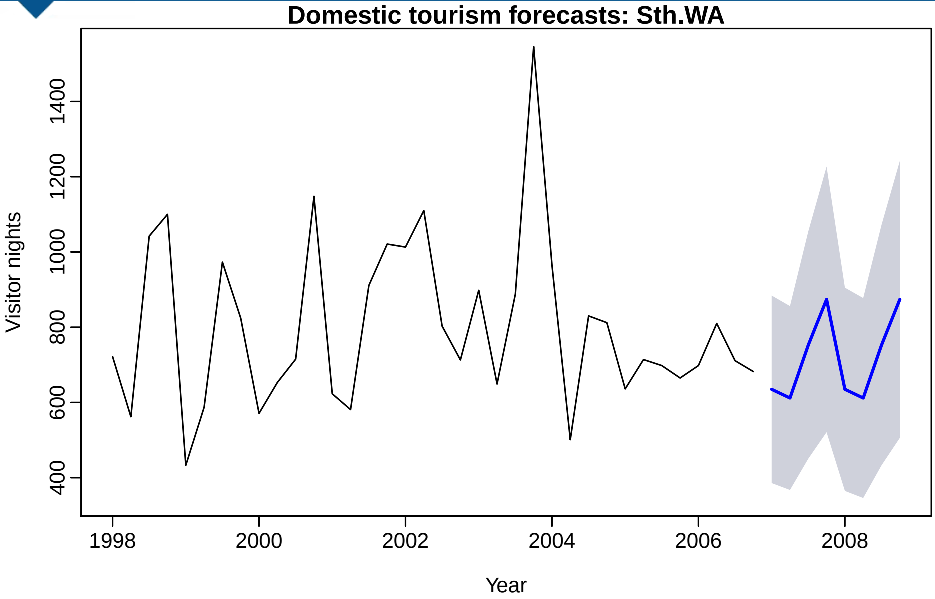
Base forecasts



Base forecasts



Base forecasts



Forecast evaluation

- Select models using all observations;
- Re-estimate models using first 12 observations and generate 1- to 8-steps ahead forecasts;
- Increase sample size one observation at a time, re-estimate models, generate forecasts until the end of the sample;
- In total 24 1-step-ahead, 23 2-steps-ahead, up to 17 8-steps-ahead for forecast evaluation.

Forecast evaluation

- Select models using all observations;
- Re-estimate models using first 12 observations and generate 1- to 8-steps ahead forecasts;
- Increase sample size one observation at a time, re-estimate models, generate forecasts until the end of the sample;
- In total 24 1-step-ahead, 23 2-steps-ahead, up to 17 8-steps-ahead for forecast evaluation.

Forecast evaluation

- Select models using all observations;
- Re-estimate models using first 12 observations and generate 1- to 8-steps ahead forecasts;
- Increase sample size one observation at a time, re-estimate models, generate forecasts until the end of the sample;
- In total 24 1-step-ahead, 23 2-steps-ahead, up to 17 8-steps-ahead for forecast evaluation.

Forecast evaluation

- Select models using all observations;
- Re-estimate models using first 12 observations and generate 1- to 8-steps ahead forecasts;
- Increase sample size one observation at a time, re-estimate models, generate forecasts until the end of the sample;
- In total 24 1-step-ahead, 23 2-steps-ahead, up to 17 8-steps-ahead for forecast evaluation.

Hierarchy: states, zones, regions

MAPE	1	2	4	6	8	Average
	Forecast Horizon (h)					
			Top Level: Australia			
Bottom-up	3.79	3.58	4.01	4.55	4.24	4.06
OLS	3.83	3.66	**3.88**	**4.19**	4.25	**3.94**
Scaling (st. dev.)	**3.68**	**3.56**	3.97	4.57	4.25	4.04
Scaling (indep.)	3.76	3.60	4.01	4.58	**4.22**	4.06
			Level 1: States			
Bottom-up	10.70	10.52	10.85	11.46	11.27	11.03
OLS	11.07	10.58	11.13	11.62	12.21	11.35
Scaling (st. dev.)	**10.44**	**10.17**	**10.47**	**10.97**	**10.98**	**10.67**
Scaling (indep.)	10.59	10.36	10.69	11.27	11.21	10.89

Hierarchy: states, zones, regions

MAPE	*Forecast Horizon (h)* 1	2	4	6	8	Average
			Level 2: Zones			
Bottom-up	14.99	14.97	14.98	15.69	15.65	15.32
OLS	15.16	15.06	15.27	15.74	16.15	15.48
Scaling (st. dev.)	**14.63**	**14.62**	**14.68**	**15.17**	**15.25**	**14.94**
Scaling (indep.)	14.79	14.79	14.85	15.46	15.49	15.14
			Bottom Level: Regions			
Bottom-up	33.12	32.54	32.26	33.74	33.96	33.18
OLS	35.89	33.86	34.26	36.06	37.49	35.43
Scaling (st. dev.)	**31.68**	**31.22**	**31.08**	**32.41**	**32.77**	**31.89**
Scaling (indep.)	32.84	32.20	32.06	33.44	34.04	32.96

Outline

1 Hierarchical and grouped time series

2 Optimal forecasts

3 Approximately optimal forecasts

4 Fast computation

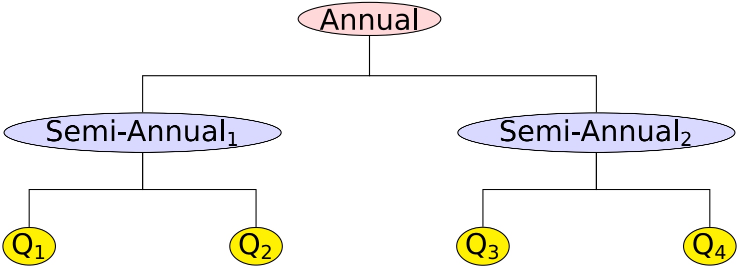
5 Example: Australian tourism

6 **Example: Temporal hierarchies**

7 hts package for R

8 References

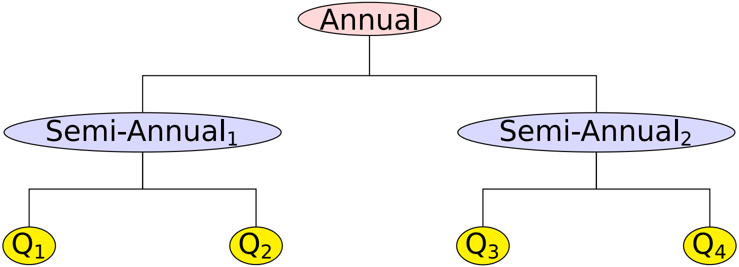
Temporal hierarchies



Basic idea:

- Forecast series at each available frequency.
- Optimally combine forecasts within the same year.

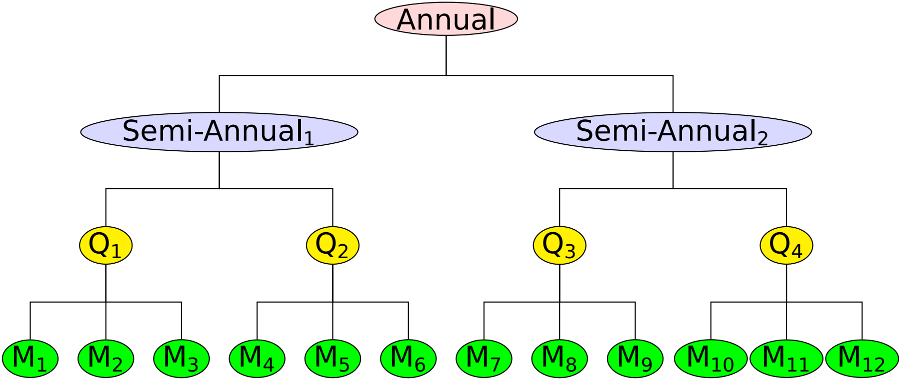
Temporal hierarchies



Basic idea:

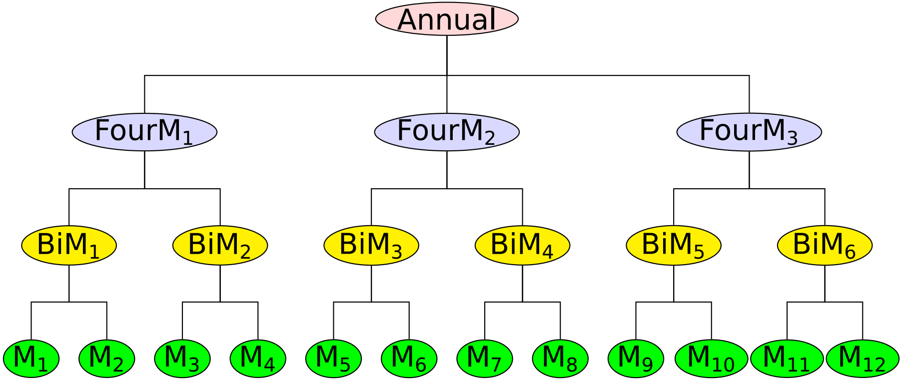
- Forecast series at each available frequency.
- Optimally combine forecasts within the same year.

Monthly series



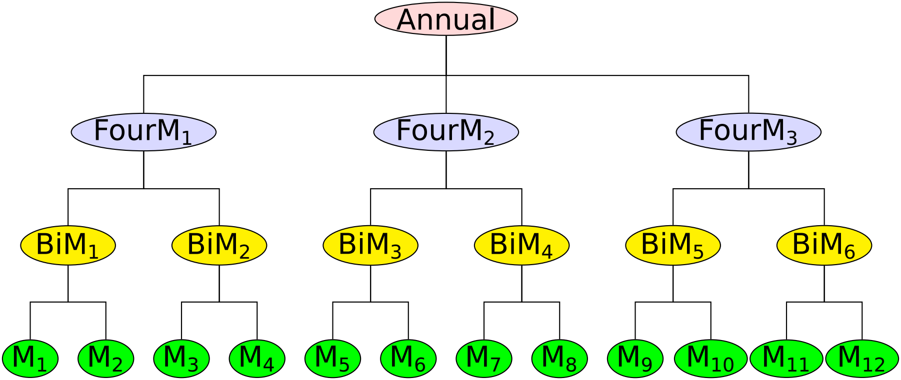
- $k = 2, 4, 12.$
- Alternatively $k = 3, 6, 12.$
- How about: $k = 2, 3, 4, 6, 12?$

Monthly series



- $k = 2, 4, 12.$
- Alternatively $k = 3, 6, 12.$
- How about: $k = 2, 3, 4, 6, 12?$

Monthly series



- $k = 2, 4, 12.$
- Alternatively $k = 3, 6, 12.$
- How about: $k = 2, 3, 4, 6, 12?$

Monthly data

$$
\underbrace{\begin{pmatrix} A \\ SemiA_1 \\ SemiA_2 \\ FourM_1 \\ FourM_2 \\ FourM_3 \\ Q_1 \\ \vdots \\ Q_4 \\ BiM_1 \\ \vdots \\ BiM_6 \\ M_1 \\ \vdots \\ M_{12} \end{pmatrix}}_{(28\times 1)} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & \vdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & \vdots & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ & & & & & & \boldsymbol{I}_{12} & & & & & \end{pmatrix}}_{\boldsymbol{S}} \underbrace{\begin{pmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \\ M_7 \\ M_8 \\ M_9 \\ M_{10} \\ M_{11} \\ M_{12} \end{pmatrix}}_{\boldsymbol{B}_t}
$$

In general

- For a time series $\{y_1, \dots, y_n\}$ observed at highest available frequency, we generate aggregate series

$$y_i^{[k]} = \sum_{t=1+(i-1)k}^{ik} y_t$$

for $i = 1, \dots, \lfloor n/k \rfloor$.

- For quarterly series: $k = 2, 4$.
- Remove $n - \lfloor n/k \rfloor$ observations from beginning of sample.

Experimental setup

- M3 forecasting competition (Makridakis and Hibon, 2000, *IJF*).
- 3003 series in total.
- 1428 monthly series with a test sample of 12 observations each.
- 756 quarterly series with a test sample of 8 observations each.

Results: Monthly

MAPE *(obs)*	*Forecast Horizon (h)* Annual (1)	SemiA (2)	FourM (3)	Q (4)	BiM (6)	M (12)	Average
			ETS				
Initial	9.66	9.18	9.76	10.14	10.82	12.85	10.40
Bottom-up	8.38	9.14	9.78	10.06	11.04	12.85	10.21
OLS	7.80	8.64	9.39	9.72	10.68	12.68	9.82
Scaling	7.64	8.44	9.15	9.49	10.45	12.40	9.60
Averaging	**7.51**	**8.31**	**9.05**	**9.38**	**10.34**	**12.30**	**9.48**

Results: Quarterly

MAPE *(obs)*	*Forecast Horizon (h)* Annual *(2)*	Semi-Ann *(4)*	Quart *(8)*	Average
		ETS		
Initial	10.50	9.97	9.84	10.10
Bottom-up	8.87	9.35	9.84	9.35
OLS	9.31	9.78	10.28	9.79
Scaling	**8.75**	**9.19**	**9.70**	**9.21**
Averaging	8.81	9.26	9.78	9.28

Outline

1 Hierarchical and grouped time series

2 Optimal forecasts

3 Approximately optimal forecasts

4 Fast computation

5 Example: Australian tourism

6 Example: Temporal hierarchies

7 hts package for R

8 References

hts package for R

hts: Hierarchical and grouped time series

Methods for analysing and forecasting hierarchical and grouped time series

Version:	4.2
Depends:	forecast ($\geq$ 5.0)
Imports:	SparseM, parallel
Published:	2014-04-09
Author:	Rob J Hyndman, Earo Wang and Alan Lee
Maintainer:	Rob J Hyndman <Rob.Hyndman at monash.edu>
License:	GPL-2 $\mid$ GPL-3 [expanded from: GPL ($\geq$ 2)]

Example using R

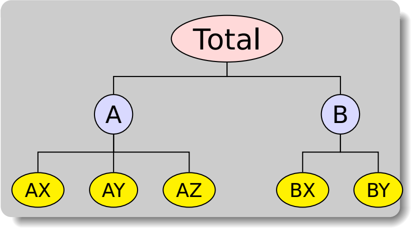
```r
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))
```

Example using R

```
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))
```



Example using R

```r
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))

# Forecast 10-step-ahead using OLS combination method
# ETS used for each series by default
fc <- forecast(y, h=10)
```

Example using R

```r
library(hts)

# bts is a matrix containing the bottom level time series
# nodes describes the hierarchical structure
y <- hts(bts, nodes=list(2, c(3,2)))

# Forecast 10-step-ahead using OLS combination method
# ETS used for each series by default
fc <- forecast(y, h=10)

# Select your own methods
ally <- aggts(y)
allf <- matrix(, nrow=10, ncol=ncol(ally))
for(i in 1:ncol(ally))
  allf[,i] <- mymethod(ally[,i], h=10)
allf <- ts(allf, start=2004)
# Reconcile forecasts so they add up
fc2 <- combinef(allf, nodes=y$nodes)
```

hts function

Usage

```
hts(y, nodes)
gts(y, groups)
```

Arguments

`y`	Multivariate time series containing the bottom level series
`nodes`	List giving number of child nodes for each level except last
`groups`	Group matrix indicating the group structure, with one column for each series when completely disaggregated, and one row for each grouping of the time series.

forecast.gts function

Usage

```
forecast(object, h,
  method = c("comb", "bu", "mo", "tdgsf", "tdgsa", "tdfp"),
  fmethod = c("ets", "rw", "arima"),
  weights = c("none", "sd", "nseries"),
  xreg = NULL, newxreg = NULL, ...)
```

Arguments

`object`	Hierarchical time series object of class `gts`.
`h`	Forecast horizon
`method`	Method for distributing forecasts within the hierarchy.
`fmethod`	Forecasting method to use
`level`	Level used for "middle-out" method (when `method="mo"`)
`positive`	If TRUE, forecasts are forced to be strictly positive
`xreg`	When `fmethod = "arima"`, a vector or matrix of external regressors, which must have the same number of rows as the original univariate time series
`newxreg`	When `fmethod = "arima"`, a vector or matrix of external regressors, which must have the same number of rows as the original univariate time series

Utility functions

`aggts(y)`	Returns time series from selected levels.
`smatrix(y)`	Returns the summing matrix
`combinef(f)`	Combines initial forecasts optimally.

More information

hts: An R Package for Forecasting Hierarchical or Grouped Time Series

Rob J Hyndman, George Athanasopoulos, Han Lin Shang

Abstract

Vignette on CRAN

This paper describes several methods that are curr
for forecasting hierarchical time series. The methods included are: top-down, bottom-up, middle-out and optimal combination. The implementation of these methods is illustrated by using regional infant mortality counts in Australia.

Keywords: top-down, bottom-up, middle-out, optimal combination .

Introduction

Advances in data collection and storage have resulted in large numbers of time series that are hierarchical in structure, and clusters of which may be correlated. In many applications the

Outline

1 Hierarchical and grouped time series

2 Optimal forecasts

3 Approximately optimal forecasts

4 Fast computation

5 Example: Australian tourism

6 Example: Temporal hierarchies

7 hts package for R

8 **References**

References

RJ Hyndman, RA Ahmed, G Athanasopoulos, and HL Shang (2011). "Optimal combination forecasts for hierarchical time series". *Computational Statistics and Data Analysis* **55**(9), 2579–2589

RJ Hyndman, E Wang, and A Lee (2014). *hts: Hierarchical time series*. `cran.r-project.org/package=hts`.

RJ Hyndman and G Athanasopoulos (2014). *Forecasting: principles and practice*. OTexts. `OTexts.org/fpp/`.

References

RJ Hyndman, RA Ahmed, G Athanasopoulos, and HL Shang (2011). "Optimal combination forecasts for hierarchical time series". *Computational Statistics and Data Analysis* **55**(9), 2579–2589

RJ Hyndman, E Wang, and A Lee (2014). *hts: Hierarchical time series*.

- ➥ Papers and R code: **robjhyndman.com**
- ➥ Email: **Rob.Hyndman@monash.edu**