Rob J Hyndman

Forecasting hierarchical time series
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
Examples

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

Hierarchical and grouped time series
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

A

Alimentary tract and metabolism

84 classes

A10

Drugs used in diabetes

A10B

Blood glucose lowering drugs

A10BA

Biguanides

A10BA02

Metformin
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.
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.
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.
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 \( t \).

\[ B_t : \] vector of all series at bottom level in time \( t \).
Hierarchical data

\[ Y_t : \text{observed aggregate of all series at time } t. \]
\[ Y_{X,t} : \text{observation on series } X \text{ at time } t. \]
\[ B_t : \text{vector of all series at bottom level in time } t. \]
Hierarchical data

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

- \( Y_t \): observed aggregate of all series at time \( t \).
- \( Y_{X,t} \): observation on series \( X \) at time \( t \).
- \( B_t \): vector of all series at bottom level in time \( t \).
Hierarchical data

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

- \( Y_t \): observed aggregate of all series at time \( t \).
- \( Y_{X,t} \): observation on series \( X \) at time \( t \).
- \( B_t \): vector of all series at bottom level in time \( t \).
Hierarchical data

\[ \mathbf{Y}_t = [\mathbf{Y}_t, \mathbf{Y}_{A,t}, \mathbf{Y}_{B,t}, \mathbf{Y}_{C,t}]' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{Y}_{A,t} \\ \mathbf{Y}_{B,t} \\ \mathbf{Y}_{C,t} \end{pmatrix} \]

\( \mathbf{Y}_t \) : observed aggregate of all series at time \( t \).
\( \mathbf{Y}_{X,t} \) : observation on series \( X \) at time \( t \).
\( \mathbf{B}_t \) : vector of all series at bottom level in time \( t \).
Hierarchical data

\[ Y_t : \text{observed aggregate of all series at time } t. \]
\[ Y_{X,t} : \text{observation on series } X \text{ at time } t. \]
\[ B_t : \text{vector of all series at bottom level in time } t. \]

\[ Y_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix} = SB_t \]
Hierarchical data

\[ Y_t = \begin{pmatrix} Y_t \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} = B_t S \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \]
Hierarchical data

\[ Y_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \]

\[ B_t = \begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} \]
Hierarchical data

\[
Y_t = \begin{pmatrix}
Y_t \\
Y_{A,t} \\
Y_{B,t} \\
Y_{C,t} \\
Y_{AX,t} \\
Y_{AY,t} \\
Y_{AZ,t} \\
Y_{BX,t} \\
Y_{BY,t} \\
Y_{BZ,t} \\
Y_{CX,t} \\
Y_{CY,t} \\
Y_{CZ,t}
\end{pmatrix} = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \times \begin{pmatrix}
Y_{AX,t} \\
Y_{AY,t} \\
Y_{AZ,t} \\
Y_{BX,t} \\
Y_{BY,t} \\
Y_{BZ,t} \\
Y_{CX,t} \\
Y_{CY,t} \\
Y_{CZ,t}
\end{pmatrix} = SB_t
\]
### Grouped data

<table>
<thead>
<tr>
<th>AX</th>
<th>AY</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>BX</td>
<td>BY</td>
<td>B</td>
</tr>
<tr>
<td>X</td>
<td>Y</td>
<td>Total</td>
</tr>
</tbody>
</table>

\[
Y_t = \begin{pmatrix}
Y_t \\
Y_{A,t} \\
Y_{B,t} \\
Y_{X,t} \\
Y_{Y,t} \\
Y_{AX,t} \\
Y_{AY,t} \\
Y_{BX,t} \\
Y_{BY,t}
\end{pmatrix} = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
\end{pmatrix} \cdot \begin{pmatrix}
Y_{AX,t} \\
Y_{AY,t} \\
Y_{BX,t} \\
Y_{BY,t}
\end{pmatrix} = S \cdot B_t
\]
### Grouped data

Forecasting hierarchical and grouped time series

\[
Y_t = \begin{pmatrix}
Y_t \\
Y_{A,t} \\
Y_{B,t} \\
Y_{X,t} \\
Y_{Y,t} \\
Y_{AX,t} \\
Y_{AY,t} \\
Y_{BX,t} \\
Y_{BY,t}
\end{pmatrix} = \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
Y_{AX,t} \\
Y_{AY,t} \\
Y_{BX,t} \\
Y_{BY,t}
\end{pmatrix}
\]

\(S\)
Grouped data

\[ Y_t = S B_t \]

Forecasting hierarchical and grouped time series

Hierarchical and grouped time series
Forecasts

Key idea: forecast reconciliation

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

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

\[
Y_t = SB_t.
\]

So \( \hat{Y}_n(h) = S\beta_n(h) + \varepsilon_h \).
Forecasts

**Key idea: forecast reconciliation**
- Ignore structural constraints and forecast every series of interest independently.
- Adjust forecasts to impose constraints.

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

\[ \mathbf{Y}_t = \mathbf{S} \mathbf{B}_t. \]

So
\[ \hat{\mathbf{Y}}_n(h) = \mathbf{S} \hat{\mathbf{\beta}}_n(h) + \mathbf{\varepsilon}_h. \]
Forecasts

Key idea: forecast reconciliation

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

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

$Y_t = SB_t$. So $\hat{Y}_n(h) = S\beta_n(h) + \varepsilon_h$.

$\beta_n(h) = E[B_{n+h} | Y_1, \ldots, Y_n].$

$\varepsilon_h$ has zero mean and covariance $\Sigma_h$.

Estimate $\beta_n(h)$ using GLS?
Key idea: forecast reconciliation

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

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

\[ Y_t = S B_t. \]

So

\[ \hat{Y}_n(h) = S \beta_n(h) + \varepsilon_h. \]

- $\beta_n(h) = E[B_{n+h} | Y_1, \ldots, Y_n]$.
- $\varepsilon_h$ has zero mean and covariance $\Sigma_h$.
- 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 $\hat{Y}_n(h)$ be vector of initial $h$-step forecasts, made at time $n$, stacked in same order as $Y_t$.

$Y_t = SB_t$. So $\hat{Y}_n(h) = S\beta_n(h) + \varepsilon_h$.

- $\beta_n(h) = \mathbb{E}[B_{n+h} \mid Y_1, \ldots, Y_n]$.
- $\varepsilon_h$ has zero mean and covariance $\Sigma_h$.
- 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 \( \hat{Y}_n(h) \) be vector of initial \( h \)-step forecasts, made at time \( n \), stacked in same order as \( Y_t \).

\[
Y_t = S B_t.
\]

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

- \( \beta_n(h) = \mathbb{E}[B_{n+h} \mid Y_1, \ldots, Y_n] \).
- \( \varepsilon_h \) has zero mean and covariance \( \Sigma_h \).
- 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 \( \hat{Y}_n(h) \) be vector of initial \( h \)-step forecasts, made at time \( n \), stacked in same order as \( Y_t \).

\[
Y_t = SB_t.
\]

So

\[
\hat{Y}_n(h) = S\beta_n(h) + \varepsilon_h.
\]

- \( \beta_n(h) = E[B_{n+h} \mid Y_1, \ldots, Y_n] \).
- \( \varepsilon_h \) has zero mean and covariance \( \Sigma_h \).
- Estimate \( \beta_n(h) \) using GLS?
Optimal combination forecasts

\[ \tilde{Y}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h S)^{-1}S'\Sigma_h \hat{Y}_n(h) \]

\[ \Sigma_h \] is generalized inverse of \( \Sigma_h \).

\[ \text{Var}[\tilde{Y}_n(h)|Y_1, \ldots, Y_n] = S(S'\Sigma_h S)^{-1}S' \]

\( \Sigma_h \) hard to estimate.
\[ \hat{\mathbf{Y}}_n(h) = \mathbf{s}\hat{\boldsymbol{\beta}}_n(h) = \mathbf{s}(\mathbf{s}'\Sigma_h\mathbf{s})^{-1}\mathbf{s}'\Sigma_h\hat{\mathbf{Y}}_n(h) \]

Initial forecasts

\[ \text{Var}[\hat{\mathbf{Y}}_n(h)|\mathbf{Y}_1, \ldots, \mathbf{Y}_n] = \mathbf{s}(\mathbf{s}'\Sigma_h\mathbf{s})^{-1}\mathbf{s}'\Sigma_h, \]

\( \Sigma_h \) is generalized inverse of \( \Sigma_h \).

Problem: \( \Sigma_h \) hard to estimate.
Optimal combination forecasts

\[
\tilde{Y}_n(h) = S\hat{\beta}_n(h) = S(S'\Sigma_h\Sigma_h^+)^{-1}S'\Sigma_h^+\hat{Y}_n(h)
\]

Revised forecasts

Initial forecasts

\[\Sigma_h^+ \text{ is generalized inverse of } \Sigma_h.\]

\[\text{Var} [\tilde{Y}_n(h) | Y_1, \ldots, Y_n] = S(S'\Sigma_h^+S)^{-1}S'\]

Problem: \(\Sigma_h\) hard to estimate.
Optimal combination forecasts

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

Revised forecasts

- \( \Sigma_h^\dagger \) is generalized inverse of \( \Sigma_h \).

- \( \text{Var}[\tilde{Y}_n(h)|Y_1, \ldots, Y_n] = S(S'\Sigma_h^\dagger S)^{-1}S' \)

Initial forecasts

**Problem:** \( \Sigma_h \) hard to estimate.
Optimal combination forecasts

\[ \tilde{Y}_n(h) = s\hat{\beta}_n(h) = s(s'\Sigma_h^\dagger s)^{-1}s'\Sigma_h^\dagger \hat{Y}_n(h) \]

Revised forecasts \hspace{2cm} Initial forecasts

- \( \Sigma_h^\dagger \) is generalized inverse of \( \Sigma_h \).
- \( \text{Var}[\tilde{Y}_n(h)|Y_1, \ldots, Y_n] = s(s'\Sigma_h^\dagger s)^{-1}s' \)
- **Problem:** \( \Sigma_h \) hard to estimate.
Optimal combination forecasts

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

Revised forecasts

\[ \text{Problem: } \Sigma_h \text{ hard to estimate.} \]

- \( \Sigma_h^\dagger \) is generalized inverse of \( \Sigma_h \).
- \( \text{Var}[\tilde{Y}_n(h)|Y_1, \ldots, Y_n] = S(S' \Sigma_h^\dagger S)^{-1} S' \)
Optimal combination forecasts

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

Revised forecasts

Initial forecasts

- $\Sigma_h^\dagger$ is generalized inverse of $\Sigma_h$.
- $\text{Var}[\tilde{Y}_n(h)|Y_1, \ldots, Y_n] = S(S'\Sigma_h^\dagger S)^{-1}S'$

**Problem:** $\Sigma_h$ hard to estimate.
Optimal combination forecasts

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

Revised forecasts \hspace{1cm} Base forecasts

Solution 1: OLS

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

\[
\tilde{Y}_n(h) = S(S'S)^{-1}S'\hat{Y}_n(h)
\]
Optimal combination forecasts

\[ \tilde{Y}_n(h) = s(s'\Sigma_h^\dagger s)^{-1}s'\Sigma_h^\dagger \hat{Y}_n(h) \]

Revised forecasts

Base forecasts

Solution 1: OLS

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

\[ \tilde{Y}_n(h) = s(s's)^{-1}s'\hat{Y}_n(h) \]
Optimal combination forecasts

\[ \tilde{Y}_n(h) = s(s' \Sigma_h s)^{-1} s' \Sigma_h \hat{Y}_n(h) \]

Revised forecasts \hspace{1cm} Base forecasts

**Solution 1: OLS**

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

\[ \tilde{Y}_n(h) = s(s's)^{-1} s' \hat{Y}_n(h) \]
Optimal combination forecasts

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

Revised forecasts

Base forecasts

Solution 1: OLS

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

\[ \tilde{Y}_n(h) = S(S'S)^{-1}S'\hat{Y}_n(h) \]
Optimal combination forecasts

\[ \tilde{Y}_n(h) = S (S' \Sigma_h S)^{-1} S' \Sigma_h \hat{Y}_n(h) \]

Revised forecasts

Base forecasts

**Solution 1: OLS**

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

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

Forecasting hierarchical and grouped time series

Approximately optimal forecasts
Optimal combination forecasts

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

Revised forecasts

Base forecasts

Solution 1: OLS

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

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

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

\[ \hat{Y}_n(h) = S(S'S)^{-1}S'\hat{Y}_n(h) \]

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

\[ \hat{Y}_n(h) = S(S'S)^{-1}S'\hat{Y}_n(h) \]

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

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

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

\[ \tilde{Y}_n(h) = S(S'S)^{-1}S'\hat{Y}_n(h) \]
Optimal combination forecasts

\[ \hat{Y}_n(h) = S(S'S)^{-1}S'\hat{Y}_n(h) \]

Weights:

\[ s(S'S)^{-1}s' = \begin{bmatrix} 0.75 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.75 & -0.25 & -0.25 \\ 0.25 & -0.25 & 0.75 & -0.25 \\ 0.25 & -0.25 & -0.25 & 0.75 \end{bmatrix} \]
Optimal combination forecasts

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

\[
\begin{bmatrix}
0.69 & 0.23 & 0.23 & 0.23 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 & 0.08 \\
0.23 & 0.58 & -0.17 & -0.17 & 0.19 & 0.19 & 0.19 & -0.06 & -0.06 & -0.06 & -0.06 & -0.06 \\
0.23 & -0.17 & 0.58 & -0.17 & -0.06 & -0.06 & -0.06 & 0.19 & 0.19 & 0.19 & -0.06 & -0.06 \\
0.23 & -0.17 & -0.17 & 0.58 & -0.06 & -0.06 & -0.06 & -0.06 & 0.19 & 0.19 & 0.19 & 0.19 \\
0.08 & 0.19 & -0.06 & -0.06 & 0.73 & -0.27 & -0.27 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
0.08 & 0.19 & -0.06 & -0.06 & -0.27 & 0.73 & -0.27 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
0.08 & 0.19 & -0.06 & -0.06 & -0.27 & -0.27 & 0.73 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & 0.73 & -0.27 & -0.27 & -0.02 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & -0.02 & 0.73 & -0.27 & -0.02 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & 0.73 & -0.27 & -0.02 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & 0.73 & -0.27 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & 0.73 \\
0.08 & -0.06 & 0.19 & -0.06 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 & -0.02 \\
\end{bmatrix}
\]
Optimal combination forecasts

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

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

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

\[
\hat{Y}_n(h) = S(S'S)^{-1}S'\hat{Y}_n(h)
\]
Challenges

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

\[ \hat{\mathbf{Y}}_n(h) = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\hat{\mathbf{Y}}_n(h) \]
Challenges

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

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

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

\[ \hat{Y}_n(h) = S(S' S)^{-1} S' \hat{Y}_n(h) \]
Optimal combination forecasts

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

Solution 1: OLS

- Approximate $\Sigma_1^\dagger$ by $cI$.

Solution 2: Rescaling

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

$$\tilde{Y}_n(h) = S(S'\Lambda S)^{-1}S'\Lambda \hat{Y}_n(h)$$
Solution 1: OLS
- Approximate $\Sigma_1^{\dagger}$ by $cI$.

Solution 2: Rescaling
- Suppose we approximate $\Sigma_1$ by its diagonal.
- Let $\Lambda = \left[\text{diagonal}(\Sigma_1)\right]^{-1}$ contain inverse one-step forecast variances.

\[
\hat{Y}_n(h) = S(S'\Lambda S)^{-1}S'\Lambda \hat{Y}_n(h)
\]
Optimal combination forecasts

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

Solution 1: OLS
- Approximate \( \Sigma_1^\dagger \) by \( cI \).

Solution 2: Rescaling
- Suppose we approximate \( \Sigma_1 \) by its diagonal.
- Let \( \Lambda = \left[ \text{diagonal}(\Sigma_1) \right]^{-1} \) contain inverse one-step forecast variances.

\[ \tilde{Y}_n(h) = S(S'\Lambda S)^{-1}S'\Lambda \hat{Y}_n(h) \]
Optimal combination forecasts

Solution 1: OLS
- Approximate $\Sigma_1^\dagger$ by $cI$.

Solution 2: Rescaling
- Suppose we approximate $\Sigma_1$ by its diagonal.
- Let $\Lambda = \left[\text{diagonal}(\Sigma_1)\right]^{-1}$ contain inverse one-step forecast variances.
Optimal combination forecasts

\[ \tilde{Y}_n(h) = S(S'\Sigma_1^+S)^{-1}S'\Sigma_1^+\hat{Y}_n(h) \]

Solution 1: OLS

- Approximate \( \Sigma_1^+ \) by \( cI \).

Solution 2: Rescaling

- Suppose we approximate \( \Sigma_1 \) by its diagonal.

- Let \( \Lambda = [\text{diagonal}(\Sigma_1)]^{-1} \) contain inverse one-step forecast variances.

\[ \tilde{Y}_n(h) = S(S'\Lambda S)^{-1}S'\Lambda \hat{Y}_n(h) \]
Solution 3: Averaging

- If the bottom level error series are approximately uncorrelated and have similar variances, then $\Lambda$ is inversely proportional to the number of series contributing to each node.

- So set $\Lambda$ to be the inverse row sums of $S$. 

\[ \hat{Y}_n(h) = S (S' \Lambda S)^{-1} S' \Lambda \hat{Y}_n(h) \]
Solution 3: Averaging

- If the bottom level error series are approximately uncorrelated and have similar variances, then $\Lambda$ is inversely proportional to the number of series contributing to each node.

- So set $\Lambda$ to be the inverse row sums of $S$. 

$$\tilde{Y}_n(h) = S(S'\Lambda S)^{-1}S'\Lambda \hat{Y}_n(h)$$
To efficiently compute the reconciled forecasts for large hierarchies or groups of time series, we must compute

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

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

**Fast computation**
Think of the hierarchy as a tree of trees:

\[ \text{Total} \]

\[ T_1 \]
\[ T_2 \]
\[ \ldots \]
\[ T_K \]

Then the summing matrix contains \( k \) smaller summing matrices:

\[
S = \begin{bmatrix}
1_{n_1}' & 1_{n_2}' & \cdots & 1_{n_K}' \\
S_1 & 0 & \cdots & 0 \\
0 & S_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & S_K
\end{bmatrix}
\]

where \( 1_n \) is an \( n \)-vector of ones and tree \( T_i \) has \( n_i \) terminal nodes.
Think of the hierarchy as a tree of trees:

![Diagram of a hierarchy with nodes labeled $T_1$, $T_2$, ..., $T_K$](image)

Then the summing matrix contains $k$ smaller summing matrices:

$$
S = \begin{bmatrix}
1'_{n_1} & 1'_{n_2} & \cdots & 1'_{n_K} \\
S_1 & 0 & \cdots & 0 \\
0 & S_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & S_K
\end{bmatrix}
$$

where $1_n$ is an $n$-vector of ones and tree $T_i$ has $n_i$ terminal nodes.
Fast computation: hierarchies

\[ s' \Lambda s = \lambda_0 J_n + \begin{bmatrix}
  s_1' \Lambda_1 s_1 & 0 & \cdots & 0 \\
  0 & s_2' \Lambda_2 s_2 & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & s_K' \Lambda_K s_K
\end{bmatrix} \]

- \( \lambda_0 \) is the top left element of \( \Lambda \);
- \( \Lambda_k \) is a block of \( \Lambda \), corresponding to tree \( T_k \);
- \( J_n \) is a matrix of ones;
- \( n = \sum_k n_k \).

Now apply the Sherman-Morrison formula ...
Fast computation: hierarchies

\[ s' \Lambda s = \lambda_0 J_n + \begin{bmatrix} s'_1 \Lambda_1 s_1 & 0 & \cdots & 0 \\ 0 & s'_2 \Lambda_2 s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s'_k \Lambda_k s_k \end{bmatrix} \]

- \( \lambda_0 \) is the top left element of \( \Lambda \);
- \( \Lambda_k \) is a block of \( \Lambda \), corresponding to tree \( T_k \);
- \( J_n \) is a matrix of ones;
- \( n = \sum_k n_k \).

Now apply the Sherman-Morrison formula . . .
Fast computation: hierarchies

\[
(S' \Lambda S)^{-1} = \begin{bmatrix}
(S_1' \Lambda_1 S_1)^{-1} & 0 & \cdots & 0 \\
0 & (S_2' \Lambda_2 S_2)^{-1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & (S_K' \Lambda_K S_K)^{-1}
\end{bmatrix} - c S_0
\]

- \( S_0 \) can be partitioned into \( K^2 \) blocks, with the \((k, \ell)\) block (of dimension \( n_k \times n_\ell \)) being

\[
(S_k' \Lambda_k S_k)^{-1} J_{n_k,n_\ell} (S_\ell' \Lambda_\ell S_\ell)^{-1}
\]

- \( J_{n_k,n_\ell} \) is a \( n_k \times n_\ell \) matrix of ones.
- \( c^{-1} = \lambda_0^{-1} + \sum_k 1'_{n_k} (S_k' \Lambda_k S_k)^{-1} 1_{n_k} \).
- Each \( S_k' \Lambda_k S_k \) can be inverted similarly.
- \( S' \Lambda Y \) can also be computed recursively.
Fast computation: hierarchies

\[
(S' \Lambda S)^{-1} = 
\begin{bmatrix}
(S'_1 \Lambda_1 S_1)^{-1} & 0 & \cdots & 0 \\
0 & (S'_2 \Lambda_2 S_2)^{-1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & (S'_K \Lambda_K S_K)^{-1}
\end{bmatrix}
- c S_0
\]

- \( S_0 \) can be partitioned into \( K^2 \) blocks, with the \((k, \ell)\) block (of dimension \( n_k \times n_\ell \)) being

- \( J_{n_k,n_\ell} \)
- \( c^{-1} \)

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

- Each \( S'_k \Lambda_k S_k \) can be inverted similarly.
- \( S' \Lambda Y \) can also be computed recursively.
Fast computation: grouped data

\[ Y_t = SB_t \]

Forecasting hierarchical and grouped time series

Fast computation

29
Fast computation: grouped data

\[ Y_t = S B_t \]

\[ Y_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{X,t} \\ Y_{Y,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Y_{AX,t} \\ Y_{AY,t} \\ Y_{BX,t} \\ Y_{BY,t} \end{pmatrix} \]
Fast computation: grouped data

\[ S = \begin{bmatrix}
1'_m \otimes 1'_n \\
1'_m \otimes I_n \\
1_m \otimes 1'_n \\
1_m \otimes I_n
\end{bmatrix} \]

\[ S' \Lambda S = \lambda_{00} J_{mn} + (\Lambda_R \otimes J_n) + (J_m \otimes \Lambda_C) + \Lambda_U \]

- \( \Lambda_R, \Lambda_C \) and \( \Lambda_U \) are diagonal matrices corresponding to rows, columns and unaggregated series;
- \( \lambda_{00} \) corresponds to aggregate.
Fast computation: grouped data

\[
S = \begin{bmatrix}
1'_m \otimes 1'_n \\
1'_m \otimes I_n \\
I_m \otimes 1'_n \\
I_m \otimes I_n
\end{bmatrix}
\]

\[
S' \Lambda S = \lambda_{00} J_{mn} + (\Lambda_R \otimes J_n) + (J_m \otimes \Lambda_C) + \Lambda_U
\]

- \(\Lambda_R, \Lambda_C\) and \(\Lambda_U\) are diagonal matrices corresponding to rows, columns and unaggregated series;
- \(\lambda_{00}\) corresponds to aggregate.
Fast computation: grouped data

\[(SΛS)^{-1} = A - \frac{A1_{mn}1'_{mn}A}{1/\lambda_{00} + 1'_{mn}A1_{mn}}\]

\[A = \Lambda_U^{-1} - \Lambda_U^{-1}(J_m \otimes D)\Lambda_U^{-1} - EM^{-1}E'.\]

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

\(E\) has \(m \times m\) blocks where \(e_{ij}\) has \(k\)th element

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

\(M\) is \(m \times m\) with \((i, j)\) element

\[
(M)_{ij} = \begin{cases} 
1 + \lambda_{i0} \sum_k \lambda_{ik}^{-1} - \lambda_{i0} \sum_k \lambda_{ik}^{-2} d_k, & i = j, \\
-\lambda_{i0}^{1/2} \lambda_{j0}^{1/2} \sum_k \lambda_{ik}^{\-1} \lambda_{jk}^{\-1} d_k, & i \neq j.
\end{cases}
\]
Fast computation: grouped data

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

\[
A = \Lambda_U^{-1} - \Lambda_U^{-1} (J_m \otimes D) \Lambda_U^{-1} - EM^{-1} E'.
\]

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

\(E\) has \(m \times m\) blocks where \(e_{ij}\) has \(k\)th element \(\{\lambda_1/2 \lambda_{0j} \sum_k \lambda_{ik}^{-1} - \lambda_{0j} \sum_k \lambda_{ik}^{-2} d_k, \ i = j, \ -\lambda_{1/2} \lambda_{0j}^{1/2} \sum_k \lambda_{ik}^{-1} \lambda_{jk}^{-1} d_k, \ i \neq j.\)

Again, the calculations can be done in such a way that we never store any of the large matrices involved.
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
Forecasting hierarchical and grouped time series

Example: Australian tourism
Australian tourism

Domestic visitor nights

Geographical hierarchy:

- **States (7)**
- **Zones (27)**
- **Regions (82)**
Base forecasts

Domestic tourism forecasts: Total

<table>
<thead>
<tr>
<th>Year</th>
<th>Visitor nights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td></td>
</tr>
</tbody>
</table>

Year
Base forecasts

Domestic tourism forecasts: NSW

<table>
<thead>
<tr>
<th>Year</th>
<th>Visitor nights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1998</td>
<td>18000</td>
</tr>
<tr>
<td>2000</td>
<td>22000</td>
</tr>
<tr>
<td>2002</td>
<td>26000</td>
</tr>
<tr>
<td>2004</td>
<td>30000</td>
</tr>
<tr>
<td>2006</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td></td>
</tr>
</tbody>
</table>
Base forecasts

Domestic tourism forecasts: VIC

Year
Visitor nights
10000 12000 14000 16000 18000
Base forecasts

Domestic tourism forecasts: Nth.Coast.NSW

Year
Visitor nights
5000 6000 7000 8000 9000
Base forecasts

Domestic tourism forecasts: Metro.QLD

Base forecasts

Forecasting hierarchical and grouped time series

Example: Australian tourism

Domestic tourism forecasts: Sth.WA

Visitor nights

Year


400 600 800 1000 1200 1400
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 forecasts 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 forecasts 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

### Forecasting hierarchical and grouped time series

**Example: Australian tourism**

<table>
<thead>
<tr>
<th>MAPE</th>
<th>Forecast Horizon (h)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Top Level: Australia</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom-up</td>
<td>3.79</td>
<td>3.58</td>
<td>4.01</td>
<td>4.55</td>
<td>4.24</td>
<td>4.06</td>
<td></td>
</tr>
<tr>
<td>OLS</td>
<td>3.83</td>
<td>3.66</td>
<td><strong>3.88</strong></td>
<td><strong>4.19</strong></td>
<td>4.25</td>
<td><strong>3.94</strong></td>
<td></td>
</tr>
<tr>
<td>Scaling (st. dev.)</td>
<td><strong>3.68</strong></td>
<td><strong>3.56</strong></td>
<td>3.97</td>
<td>4.57</td>
<td>4.25</td>
<td>4.04</td>
<td></td>
</tr>
<tr>
<td>Scaling (indep.)</td>
<td>3.76</td>
<td>3.60</td>
<td>4.01</td>
<td>4.58</td>
<td><strong>4.22</strong></td>
<td>4.06</td>
<td></td>
</tr>
<tr>
<td><strong>Level 1: States</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom-up</td>
<td>10.70</td>
<td>10.52</td>
<td>10.85</td>
<td>11.46</td>
<td>11.27</td>
<td>11.03</td>
<td></td>
</tr>
<tr>
<td>OLS</td>
<td>11.07</td>
<td>10.58</td>
<td>11.13</td>
<td>11.62</td>
<td>12.21</td>
<td>11.35</td>
<td></td>
</tr>
<tr>
<td>Scaling (st. dev.)</td>
<td><strong>10.44</strong></td>
<td><strong>10.17</strong></td>
<td><strong>10.47</strong></td>
<td><strong>10.97</strong></td>
<td><strong>10.98</strong></td>
<td><strong>10.67</strong></td>
<td></td>
</tr>
<tr>
<td>Scaling (indep.)</td>
<td>10.59</td>
<td>10.36</td>
<td>10.69</td>
<td>11.27</td>
<td>11.21</td>
<td>10.89</td>
<td></td>
</tr>
</tbody>
</table>
### Forecasting hierarchical and grouped time series

Example: Australian tourism

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

Forecasting hierarchical and grouped time series
**Monthly series**

- **Annual**
  - Semi-Annual\textsubscript{1}
    - Q\textsubscript{1}
      - M\textsubscript{1}
      - M\textsubscript{2}
      - M\textsubscript{3}
      - M\textsubscript{4}
      - M\textsubscript{5}
    - Q\textsubscript{2}
      - M\textsubscript{6}
      - M\textsubscript{7}
      - M\textsubscript{8}
      - M\textsubscript{9}
      - M\textsubscript{10}
  - Semi-Annual\textsubscript{2}
    - Q\textsubscript{3}
      - M\textsubscript{11}
      - M\textsubscript{12}
    - Q\textsubscript{4}

- $k = 2, 4, 12$.
- 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: $k = 2, 3, 4, 6, 12$?
Forecasting hierarchical and grouped time series

Example: Temporal hierarchies

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

Forecasting hierarchical and grouped time series

Example: Temporal hierarchies

\[
\begin{align*}
\begin{pmatrix}
A \\
SemiA_1 \\
SemiA_2 \\
FourM_1 \\
FourM_2 \\
FourM_3 \\
Q_1 \\
\vdots \\
Q_4 \\
BiM_1 \\
\vdots \\
BiM_6 \\
M_1 \\
\vdots \\
M_{12}
\end{pmatrix}
&=
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{pmatrix}
\end{align*}
\]
In general

For a time series \( \{y_1, \ldots, y_n\} \) observed at highest available frequency, we generate aggregate series

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

for \( i = 1, \ldots, \lfloor n/k \rfloor \).

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

<table>
<thead>
<tr>
<th>MAPE (obs)</th>
<th>Forecast Horizon (h)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Annual (1)</td>
<td>SemiA (2)</td>
<td>FourM (3)</td>
<td>Q (4)</td>
<td>BiM (6)</td>
<td>M (12)</td>
<td>Average</td>
</tr>
<tr>
<td>ETS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom-up</td>
<td>8.38</td>
<td>9.14</td>
<td>9.78</td>
<td>10.06</td>
<td>11.04</td>
<td>12.85</td>
<td>10.21</td>
</tr>
<tr>
<td>OLS</td>
<td>7.80</td>
<td>8.64</td>
<td>9.39</td>
<td>9.72</td>
<td>10.68</td>
<td>12.68</td>
<td>9.82</td>
</tr>
<tr>
<td>Scaling</td>
<td>7.64</td>
<td>8.44</td>
<td>9.15</td>
<td>9.49</td>
<td>10.45</td>
<td>12.40</td>
<td>9.60</td>
</tr>
<tr>
<td>Averaging</td>
<td><strong>7.51</strong></td>
<td><strong>8.31</strong></td>
<td><strong>9.05</strong></td>
<td><strong>9.38</strong></td>
<td><strong>10.34</strong></td>
<td><strong>12.30</strong></td>
<td><strong>9.48</strong></td>
</tr>
</tbody>
</table>

**Forecasting hierarchical and grouped time series**

**Example: Temporal hierarchies**
## Results: Quarterly

<table>
<thead>
<tr>
<th>MAPE (obs)</th>
<th>Forecast Horizon (h)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>10.50, 9.97, 9.84, 10.10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom-up</td>
<td>8.87, 9.35, 9.84, 9.35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLS</td>
<td>9.31, 9.78, 10.28, 9.79</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaling</td>
<td>8.75, 9.19, 9.70, 9.21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Averaging</td>
<td>8.81, 9.26, 9.78, 9.28</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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

### Example: Temporal hierarchies
hts: Hierarchical and grouped time series
Methods for analysing and forecasting hierarchical and grouped time series

Version: 4.2
Depends: forecast (≥ 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 | GPL-3 [expanded from: GPL (≥ 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)))
```
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)))
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)
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 re-
    gressors, which must have the same number of rows as the
    original univariate time series
newxreg When fmethod = "arima", a vector or matrix of external re-
    gressors, which must have the same number of rows as the
    original univariate time series
## Utility functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>aggts(y)</code></td>
<td>Returns time series from selected levels.</td>
</tr>
<tr>
<td><code>smatrix(y)</code></td>
<td>Returns the summing matrix</td>
</tr>
<tr>
<td><code>combinef(f)</code></td>
<td>Combines initial forecasts optimally.</td>
</tr>
</tbody>
</table>
hts: An R Package for Forecasting Hierarchical or Grouped Time Series

Rob J Hyndman, George Athanasopoulos, Han Lin Shang

Abstract

This paper describes several methods that are currently available 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
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, E Wang, and A Lee (2014). *hts: Hierarchical time series*. cran.r-project.org/package=hts

Papers and R code: robjhyndman.com

Email: Rob.Hyndman@monash.edu