Automatic algorithms for time series forecasting
Outline

1 Motivation

2 Exponential smoothing

3 ARIMA modelling

4 Automatic nonlinear forecasting?

5 Time series with complex seasonality

6 Hierarchical and grouped time series

7 The future of forecasting
Motivation
Motivation

Automatic algorithms for time series forecasting
Motivation

Automatic algorithms for time series forecasting
Motivation

Automatic algorithms for time series forecasting
Motivation

Automatic algorithms for time series forecasting
Motivation

1. Common in business to have over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.

Specifications

Automatic forecasting algorithms must:

- determine an appropriate time series model;
- estimate the parameters;
- compute the forecasts with prediction intervals.
Motivation

1. Common in business to have over 1000 products that need forecasting at least monthly.
2. Forecasts are often required by people who are untrained in time series analysis.

Specifications

Automatic forecasting algorithms must:
- determine an appropriate time series model;
- estimate the parameters;
- compute the forecasts with prediction intervals.
Example: Asian sheep

Numbers of sheep in Asia

<table>
<thead>
<tr>
<th>Year</th>
<th>Millions of Sheep</th>
</tr>
</thead>
<tbody>
<tr>
<td>1960</td>
<td>250</td>
</tr>
<tr>
<td>1970</td>
<td>300</td>
</tr>
<tr>
<td>1980</td>
<td>350</td>
</tr>
<tr>
<td>1990</td>
<td>400</td>
</tr>
<tr>
<td>2000</td>
<td>450</td>
</tr>
<tr>
<td>2010</td>
<td>500</td>
</tr>
</tbody>
</table>
Example: Asian sheep

Automatic ETS forecasts

Year
millions of sheep
250 300 350 400 450 500 550

Automatic algorithms for time series forecasting

Motivation
Example: Corticosteroid sales

Monthly corticosteroid drug sales in Australia

<table>
<thead>
<tr>
<th>Year</th>
<th>Total scripts (millions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>0.4</td>
</tr>
<tr>
<td>2000</td>
<td>0.6</td>
</tr>
<tr>
<td>2005</td>
<td>0.8</td>
</tr>
<tr>
<td>2010</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Automatic algorithms for time series forecasting
Example: Corticosteroid sales

Forecasts from ARIMA(3,1,3)(0,1,1)[12]
Outline

1 Motivation

2 Exponential smoothing

3 ARIMA modelling

4 Automatic nonlinear forecasting?

5 Time series with complex seasonality

6 Hierarchical and grouped time series

7 The future of forecasting
### Exponential smoothing methods

#### Trend Component

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td>Ad (Additive damped)</td>
<td>Ad,N</td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td>M,N</td>
</tr>
<tr>
<td>Md (Multiplicative damped)</td>
<td>Md,N</td>
</tr>
</tbody>
</table>

#### Seasonal Component

<table>
<thead>
<tr>
<th></th>
<th>N (None)</th>
<th>A (Additive)</th>
<th>M (Multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
<td></td>
</tr>
<tr>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
<td></td>
</tr>
<tr>
<td>Ad,N</td>
<td>Ad,A</td>
<td>Ad,M</td>
<td></td>
</tr>
<tr>
<td>M,N</td>
<td>M,A</td>
<td>M,M</td>
<td></td>
</tr>
<tr>
<td>Md,N</td>
<td>Md,A</td>
<td>Md,M</td>
<td></td>
</tr>
</tbody>
</table>
## Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None)</td>
<td>N,N</td>
</tr>
<tr>
<td>(Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td>(Additive damped)</td>
<td>A_d,N</td>
</tr>
<tr>
<td>(Multiplicative)</td>
<td>M,N</td>
</tr>
<tr>
<td>(Multiplicative damped)</td>
<td>M_d,N</td>
</tr>
</tbody>
</table>

- **N, N**: Simple exponential smoothing
## Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N (None)</td>
</tr>
<tr>
<td>N (None)</td>
<td>N,N</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td>A_d (Additive damped)</td>
<td>A_d,N</td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td>M,N</td>
</tr>
<tr>
<td>M_d (Multiplicative damped)</td>
<td>M_d,N</td>
</tr>
</tbody>
</table>

- **N,N:** Simple exponential smoothing
- **A,N:** Holt’s linear method
## Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
<th>N (None)</th>
<th>A (Additive)</th>
<th>M (Multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td></td>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
</tr>
<tr>
<td>A (Additive)</td>
<td></td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
</tr>
<tr>
<td>Ad (Additive damped)</td>
<td></td>
<td>Ad,N</td>
<td>Ad,A</td>
<td>Ad,M</td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td></td>
<td>M,N</td>
<td>M,A</td>
<td>M,M</td>
</tr>
<tr>
<td>Md (Multiplicative damped)</td>
<td></td>
<td>Md,N</td>
<td>Md,A</td>
<td>Md,M</td>
</tr>
</tbody>
</table>

- **N,N**: Simple exponential smoothing
- **A,N**: Holt’s linear method
- **Ad,N**: Additive damped trend method
### Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td>Ad (Additive damped)</td>
<td>Ad,N</td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td>M,N</td>
</tr>
<tr>
<td>Md (Multiplicative damped)</td>
<td>Md,N</td>
</tr>
</tbody>
</table>

#### N,N: Simple exponential smoothing

#### A,N: Holt’s linear method

#### Ad,N: Additive damped trend method

#### M,N: Exponential trend method
### Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>N</strong> (None)</td>
<td>N,N</td>
</tr>
<tr>
<td><strong>A</strong> (Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td><strong>Ad</strong> (Additive damped)</td>
<td>A,Ad,N</td>
</tr>
<tr>
<td><strong>M</strong> (Multiplicative)</td>
<td>M,N</td>
</tr>
<tr>
<td><strong>Md</strong> (Multiplicative damped)</td>
<td>Md,N</td>
</tr>
</tbody>
</table>

**Seasonal Component**
- **N** (None)
- **A** (Additive)
- **M** (Multiplicative)

**N,N**: Simple exponential smoothing

**A,N**: Holt’s linear method

**Ad,N**: Additive damped trend method

**M,N**: Exponential trend method

**Md,N**: Multiplicative damped trend method

---

Automatic algorithms for time series forecasting  | Exponential smoothing
# Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
<th>N (None)</th>
<th>A (Additive)</th>
<th>M (Multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td></td>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
</tr>
<tr>
<td>A (Additive)</td>
<td></td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
</tr>
<tr>
<td>A_d (Additive damped)</td>
<td></td>
<td>A_d,N</td>
<td>A_d,A</td>
<td>A_d,M</td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td></td>
<td>M,N</td>
<td>M,A</td>
<td>M,M</td>
</tr>
<tr>
<td>M_d (Multiplicative damped)</td>
<td></td>
<td>M_d,N</td>
<td>M_d,A</td>
<td>M_d,M</td>
</tr>
</tbody>
</table>

- **N,N**: Simple exponential smoothing
- **A,N**: Holt’s linear method
- **A_d,N**: Additive damped trend method
- **M,N**: Exponential trend method
- **M_d,N**: Multiplicative damped trend method
- **A,A**: Additive Holt-Winters’ method
## Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
<th>N (None)</th>
<th>A (Additive)</th>
<th>M (Multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td></td>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
</tr>
<tr>
<td>A (Additive)</td>
<td></td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
</tr>
<tr>
<td>A_d (Additive damped)</td>
<td></td>
<td>A_d,N</td>
<td>A_d,A</td>
<td>A_d,M</td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td></td>
<td>M,N</td>
<td>M,A</td>
<td>M,M</td>
</tr>
<tr>
<td>M_d (Multiplicative damped)</td>
<td></td>
<td>M_d,N</td>
<td>M_d,A</td>
<td>M_d,M</td>
</tr>
</tbody>
</table>

- N,N: Simple exponential smoothing
- A,N: Holt’s linear method
- A\_d,N: Additive damped trend method
- M,N: Exponential trend method
- M\_d,N: Multiplicative damped trend method
- A,A: Additive Holt-Winters’ method
- A,M: Multiplicative Holt-Winters’ method
There are 15 separate exp. smoothing methods.
There are 15 separate exp. smoothing methods.
Each can have an additive or multiplicative error, giving 30 separate models.
### Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td>A_d (Additive damped)</td>
<td>A_d,N</td>
</tr>
<tr>
<td>M (Multipliciative)</td>
<td>M,N</td>
</tr>
<tr>
<td>M_d (Multipliciative damped)</td>
<td>M_d,N</td>
</tr>
<tr>
<td></td>
<td>N,A</td>
</tr>
<tr>
<td></td>
<td>A,A</td>
</tr>
<tr>
<td></td>
<td>A_d,A</td>
</tr>
<tr>
<td></td>
<td>M,A</td>
</tr>
<tr>
<td></td>
<td>M_d,A</td>
</tr>
<tr>
<td></td>
<td>N,M</td>
</tr>
<tr>
<td></td>
<td>A,M</td>
</tr>
<tr>
<td></td>
<td>A_d,M</td>
</tr>
<tr>
<td></td>
<td>M,M</td>
</tr>
<tr>
<td></td>
<td>M_d,M</td>
</tr>
</tbody>
</table>

- There are 15 separate exp. smoothing methods.
- Each can have an additive or multiplicative error, giving 30 separate models.
- Only 19 models are numerically stable.
### Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N</td>
</tr>
<tr>
<td>N (None)</td>
<td>N,N</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td>A_d (Additive damped)</td>
<td>A_d,N</td>
</tr>
<tr>
<td>M (Multiplicativ</td>
<td>M,N</td>
</tr>
<tr>
<td>M_d (Multiplicativ damped)</td>
<td>M_d,N</td>
</tr>
</tbody>
</table>

- There are 15 separate exp. smoothing methods.
- Each can have an additive or multiplicative error, giving 30 separate models.
- Only 19 models are numerically stable.
- Multiplicative trend models give poor forecasts leaving 15 models.
### Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
<th>N (None)</th>
<th>A (Additive)</th>
<th>M (Multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
<td></td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
<td></td>
</tr>
<tr>
<td>Ad (Additive damped)</td>
<td>Ad,N</td>
<td>Ad,A</td>
<td>Ad,M</td>
<td></td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td>M,N</td>
<td>M,A</td>
<td>M,M</td>
<td></td>
</tr>
<tr>
<td>Md (Multiplicative damped)</td>
<td>Md,N</td>
<td>Md,A</td>
<td>Md,M</td>
<td></td>
</tr>
</tbody>
</table>

**General notation**

ETS: Exponential Smoothing

- A,N,N: Simple exponential smoothing with additive errors
- A,A,N: Holt’s linear method with additive errors
- M,N,N: Multiplicative Holt-Winters’ method with multiplicative errors

**Automatic algorithms for time series forecasting**

**Exponential smoothing**
### Exponential Smoothing Methods

#### General notation

**ETS** : Exponential Smoothing

#### Trend Component

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None)</td>
<td>N</td>
</tr>
<tr>
<td>(Additive)</td>
<td>A</td>
</tr>
<tr>
<td>(Additive damped)</td>
<td>A_d</td>
</tr>
<tr>
<td>(Multiplicative)</td>
<td>M</td>
</tr>
<tr>
<td>(Multiplicative damped)</td>
<td>M_d</td>
</tr>
</tbody>
</table>

#### Seasonal Component

<table>
<thead>
<tr>
<th></th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None)</td>
<td>N</td>
</tr>
<tr>
<td>(Additive)</td>
<td>A</td>
</tr>
<tr>
<td>(Multiplicative)</td>
<td>M</td>
</tr>
</tbody>
</table>

#### Examples:

- A,N,N: Simple exponential smoothing with additive errors
- A,A,N: Holt's linear method with additive errors
- M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

---

**Automatic algorithms for time series forecasting**

**Exponential smoothing**
## Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
<th>N (None)</th>
<th>A (Additive)</th>
<th>M (Multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
<td></td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
<td></td>
</tr>
<tr>
<td>A&lt;sub&gt;d&lt;/sub&gt; (Additive damped)</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,N</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,A</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,M</td>
<td></td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td>M,N</td>
<td>M,A</td>
<td>M,M</td>
<td></td>
</tr>
<tr>
<td>M&lt;sub&gt;d&lt;/sub&gt; (Multiplicative damped)</td>
<td>M&lt;sub&gt;d&lt;/sub&gt;,N</td>
<td>M&lt;sub&gt;d&lt;/sub&gt;,A</td>
<td>M&lt;sub&gt;d&lt;/sub&gt;,M</td>
<td></td>
</tr>
</tbody>
</table>

**General notation**  
ETS : **Exponential Smoothing**  
↑  
**Trend**

**Examples:**  
A,N,N: Simple exponential smoothing with additive errors  
A,A,N: Holt’s linear method with additive errors  
M,A,M: Multiplicative Holt-Winters’ method with multiplicative errors
## Exponential smoothing methods

### General notation

\[ \text{ETS: Exponential Smoothing} \]

### Trend Component

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td>A&lt;sub&gt;d&lt;/sub&gt; (Additive damped)</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,N</td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td>M,N</td>
</tr>
<tr>
<td>M&lt;sub&gt;d&lt;/sub&gt; (Multiplicative damped)</td>
<td>M&lt;sub&gt;d&lt;/sub&gt;,N</td>
</tr>
</tbody>
</table>

### Seasonal Component

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>N (None)</th>
<th>A (Additive)</th>
<th>M (Multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N,N</td>
<td>N,N</td>
<td>N,A</td>
<td>N,M</td>
</tr>
<tr>
<td>A,N</td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
</tr>
<tr>
<td>A&lt;sub&gt;d&lt;/sub&gt;,N</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,N</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,A</td>
<td>A&lt;sub&gt;d&lt;/sub&gt;,M</td>
</tr>
<tr>
<td>M,N</td>
<td>M,N</td>
<td>M,A</td>
<td>M,M</td>
</tr>
<tr>
<td>M&lt;sub&gt;d&lt;/sub&gt;,N</td>
<td>M&lt;sub&gt;d&lt;/sub&gt;,N</td>
<td>M&lt;sub&gt;d&lt;/sub&gt;,A</td>
<td>M&lt;sub&gt;d&lt;/sub&gt;,M</td>
</tr>
</tbody>
</table>

### Examples:

- **A,N,N**: Simple exponential smoothing with additive errors
- **A,A,N**: Holt’s linear method with additive errors
- **M,A,M**: Multiplicative Holt-Winters’ method with multiplicative errors
Exponential smoothing methods

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>(None)</td>
</tr>
<tr>
<td>N (None)</td>
<td>N,N</td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
</tr>
<tr>
<td>Ad (Additive damped)</td>
<td>Ad,N</td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td>M,N</td>
</tr>
<tr>
<td>Md (Multiplicative damped)</td>
<td>Md,N</td>
</tr>
</tbody>
</table>

General notation: E T S : Exponential Smoothing

Error Trend Seasonal

Examples:
- A,N,N: Simple exponential smoothing with additive errors
- A,A,N: Holt’s linear method with additive errors
- M,A,M: Multiplicative Holt-Winters’ method with multiplicative errors

Automatic algorithms for time series forecasting
### Exponential smoothing methods

#### General notation

**ETS**: Exponential Smoothing

- **Error**
- **Trend**
- **Seasonal**

#### Examples:
- **A,N,N**: Simple exponential smoothing with additive errors
- **A,A,N**: Holt’s linear method with additive errors
- **M,A,M**: Multiplicative Holt-Winters’ method with multiplicative errors

#### Trend Component

<table>
<thead>
<tr>
<th>Trend Component</th>
<th>Seasonal Component</th>
<th>N (None)</th>
<th>A (Additive)</th>
<th>M (Multiplicative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (None)</td>
<td>N,N</td>
<td>A,N</td>
<td>N,M</td>
<td></td>
</tr>
<tr>
<td>A (Additive)</td>
<td>A,N</td>
<td>A,A</td>
<td>A,M</td>
<td></td>
</tr>
<tr>
<td>A_d (Additive damped)</td>
<td>A_d,N</td>
<td>A_d,A</td>
<td>A_d,M</td>
<td></td>
</tr>
<tr>
<td>M (Multiplicative)</td>
<td>M,N</td>
<td>M,A</td>
<td>M,M</td>
<td></td>
</tr>
<tr>
<td>M_d (Multiplicative damped)</td>
<td>M_d,N</td>
<td>M_d,A</td>
<td>M_d,M</td>
<td></td>
</tr>
</tbody>
</table>

---

**Automatic algorithms for time series forecasting**

**Exponential smoothing**

Page 9
Exponential smoothing methods

Innovations state space models

- All ETS models can be written in innovations state space form (IJF, 2002).
- Additive and multiplicative versions give the same point forecasts but different prediction intervals.

General notation

\[ \text{ETS : Exponential Smoothing} \]

\[ \uparrow \quad \downarrow \]

Error Trend Seasonal

Examples:

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt’s linear method with additive errors

M,A,M: Multiplicative Holt-Winters’ method with multiplicative errors
ETS state space model

\[ x_t = (\text{level, slope, seasonal}) \]
ETS state space model

\[ x_t = (\text{level}, \text{slope}, \text{seasonal}) \]
ETS state space model

\[ x_{t-1}, \varepsilon_t, x_t, \varepsilon_{t+1} \rightarrow y_t \rightarrow x_t \rightarrow y_{t+1} \]

State space model

\[ x_t = (\text{level, slope, seasonal}) \]
ETS state space model

State space model

\[ x_t = (\text{level}, \text{slope}, \text{seasonal}) \]
ETS state space model

\[ x_t = (\text{level, slope, seasonal}) \]
ETS state space model

\[ x_t = (\text{level}, \text{slope}, \text{seasonal}) \]
ETS state space model

\[ x_t = (\text{level}, \text{slope}, \text{seasonal}) \]
ETS state space model

$x_{t-1}$ \rightarrow $y_t$ \rightarrow $x_t$ \rightarrow $y_{t+1}$ \rightarrow $x_{t+1}$ \rightarrow $y_{t+2}$ \rightarrow $x_{t+2}$ \rightarrow $y_{t+3}$ \rightarrow $x_{t+3}$

$x_t = (\text{level, slope, seasonal})$
ETS state space model

\[ x_t = (\text{level}, \text{slope}, \text{seasonal}) \]
ETS state space model

\[ x_t = (\text{level, slope, seasonal}) \]

State space model

\[
\begin{align*}
\mathbf{x}_{t-1} & \rightarrow y_t \\
\varepsilon_t & \rightarrow \mathbf{x}_t \\
\varepsilon_{t+1} & \rightarrow y_{t+1} \\
\mathbf{x}_{t+1} & \rightarrow y_{t+2} \\
\varepsilon_{t+2} & \rightarrow \mathbf{x}_{t+2} \\
\varepsilon_{t+3} & \rightarrow y_{t+3} \\
\varepsilon_{t+4} & \rightarrow \mathbf{x}_{t+3} \\
\end{align*}
\]

Estimation

Compute likelihood \( L \) from \( \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T \).
Optimize \( L \) wrt model parameters.
Innovations state space models

Let \( \mathbf{x}_t = (\ell_t, b_t, s_t, s_{t-1}, \ldots, s_{t-m+1}) \) and \( \mathbf{\varepsilon}_t \sim \text{N}(0, \sigma^2) \).

\[
\begin{align*}
\mathbf{y}_t &= h(\mathbf{x}_{t-1}) + k(\mathbf{x}_{t-1})\mathbf{\varepsilon}_t \\
\mathbf{x}_t &= f(\mathbf{x}_{t-1}) + g(\mathbf{x}_{t-1})\mathbf{\varepsilon}_t
\end{align*}
\]

Observation equation
State equation

Additive errors:
\[
k(\mathbf{x}_{t-1}) = 1. \quad \mathbf{y}_t = \mu_t + \mathbf{\varepsilon}_t.
\]

Multiplicative errors:
\[
k(\mathbf{x}_{t-1}) = \mu_t. \quad \mathbf{y}_t = \mu_t(1 + \mathbf{\varepsilon}_t).
\]
\[
\mathbf{\varepsilon}_t = (\mathbf{y}_t - \mu_t)/\mu_t \text{ is relative error.}
\]
Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

**Estimation**

\[
L^*(\theta, x_0) = n \log \left( \sum_{t=1}^{n} \frac{\varepsilon_t^2}{k^2(x_{t-1})} \right) + 2 \sum_{t=1}^{n} \log |k(x_{t-1})| \\
= -2 \log(\text{Likelihood}) + \text{constant}
\]
Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

Estimation

\[ L^*(\theta, x_0) = n \log \left( \sum_{t=1}^{n} \frac{\varepsilon_t^2}{k^2(x_{t-1})} \right) + 2 \sum_{t=1}^{n} \log |k(x_{t-1})| \]

\[ = -2 \log(\text{Likelihood}) + \text{constant} \]

Minimize wrt \( \theta = (\alpha, \beta, \gamma, \phi) \) and initial states \( x_0 = (x_0, x_0, x_0, x_{-1}, ..., x_{-m+1}) \)
Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

Estimation

$$L^*(\theta, x_0) = n \log \left( \sum_{t=1}^{n} \frac{\varepsilon_t^2}{k^2(x_{t-1})} \right) + 2 \sum_{t=1}^{n} \log |k(x_{t-1})|$$

$$= -2 \log(\text{Likelihood}) + \text{constant}$$

- Minimize wrt $\theta = (\alpha, \beta, \gamma, \phi)$ and initial states $x_0 = (\ell_0, b_0, s_0, s_{-1}, \ldots, s_{-m+1})$. 

Automatic algorithms for time series forecasting

Exponential smoothing
Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

**Estimation**

\[ L^*(\theta, x_0) = n \log \left( \sum_{t=1}^{n} \frac{\varepsilon_t^2}{k^2(x_{t-1})} \right) + 2 \sum_{t=1}^{n} \log |k(x_{t-1})| \]

\[ = -2 \log(\text{Likelihood}) + \text{constant} \]

- Minimize wrt \( \theta = (\alpha, \beta, \gamma, \phi) \) and initial states \( x_0 = (\ell_0, b_0, s_0, s_{-1}, \ldots, s_{-m+1}) \).
Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

**Estimation**

\[
L^*(\theta, x_0) = n \log \left( \sum_{t=1}^{n} \frac{\varepsilon_t^2}{k^2(x_{t-1})} \right) + 2 \sum_{t=1}^{n} \log |k(x_{t-1})| \\
= -2 \log(\text{Likelihood}) + \text{constant}
\]

- Minimize wrt \( \theta = (\alpha, \beta, \gamma, \phi) \) and initial states \( x_0 = (\ell_0, b_0, s_0, s_{-1}, \ldots, s_{-m+1}) \).
Innovations state space models

- All models can be written in state space form.
- Additive and multiplicative versions give same point forecasts but different prediction intervals.

**Estimation**

\[
L^* (\theta, x_0) = n \log \left( \sum_{t=1}^{n} \frac{\varepsilon_t^2}{k^2(x_{t-1})} \right) + 2 \sum_{t=1}^{n} \log |k(x_{t-1})| \\
= -2 \log(\text{Likelihood}) + \text{constant}
\]

- Minimize wrt \( \theta = (\alpha, \beta, \gamma, \phi) \) and initial states \( x_0 = (\ell_0, b_0, s_0, s_{-1}, \ldots, s_{-m+1}) \).

**Q:** How to choose between the 15 useful ETS models?
Cross-validation

Traditional evaluation

- Training data
- Test data

- time
Cross-validation

Traditional evaluation

Standard cross-validation

- Training data
- Test data
- time

Automatic algorithms for time series forecasting

Exponential smoothing
Cross-validation

Traditional evaluation

Standard cross-validation

Time series cross-validation

Automatic algorithms for time series forecasting

Exponential smoothing
Cross-validation

Traditional evaluation

Standard cross-validation

Time series cross-validation

Automatic algorithms for time series forecasting  Exponential smoothing
Cross-validation

Traditional evaluation

Standard cross-validation

Time series cross-validation

Also known as “Evaluation on a rolling forecast origin”
Akaike’s Information Criterion

\[ \text{AIC} = -2 \log(L) + 2k \]

where \( L \) is the likelihood and \( k \) is the number of estimated parameters in the model.

- This is a penalized likelihood approach.
- If \( L \) is Gaussian, then \( \text{AIC} \approx c + T \log \text{MSE} + 2k \)
  where \( c \) is a constant, \( \text{MSE} \) is from one-step forecasts on training set, and \( T \) is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as \( T \to \infty \)) to minimizing MSE from one-step forecasts on test set via time series cross-validation.
Akaike’s Information Criterion

\[ \text{AIC} = -2 \log(L) + 2k \]

where \( L \) is the likelihood and \( k \) is the number of estimated parameters in the model.

- This is a \textit{penalized likelihood} approach.
- If \( L \) is Gaussian, then \( \text{AIC} \approx c + T \log \text{MSE} + 2k \)
  where \( c \) is a constant, MSE is from one-step forecasts on \textit{training set}, and \( T \) is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as \( T \to \infty \)) to minimizing MSE from one-step forecasts on \textit{test set} via time series cross-validation.
Akaike’s Information Criterion

$$\text{AIC} = -2 \log(L) + 2k$$

where $L$ is the likelihood and $k$ is the number of estimated parameters in the model.

- This is a **penalized likelihood** approach.
- If $L$ is Gaussian, then $\text{AIC} \approx c + T \log \text{MSE} + 2k$

where $c$ is a constant, MSE is from one-step forecasts on **training set**, and $T$ is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as $T \to \infty$) to minimizing MSE from one-step forecasts on **test set** via time series cross-validation.
Akaike’s Information Criterion

\[ \text{AIC} = -2 \log(L) + 2k \]

where \( L \) is the likelihood and \( k \) is the number of estimated parameters in the model.

- This is a *penalized likelihood* approach.
- If \( L \) is Gaussian, then \( \text{AIC} \approx c + T \log \text{MSE} + 2k \)

where \( c \) is a constant, MSE is from one-step forecasts on *training set*, and \( T \) is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as \( T \to \infty \)) to minimizing MSE from one-step forecasts on *test set* via time series cross-validation.

Automatic algorithms for time series forecasting

Exponential smoothing
Akaike’s Information Criterion

\[ \text{AIC} = -2 \log(L) + 2k \]

where \( L \) is the likelihood and \( k \) is the number of estimated parameters in the model.

- This is a penalized likelihood approach.
- If \( L \) is Gaussian, then \( \text{AIC} \approx c + T \log \text{MSE} + 2k \)

where \( c \) is a constant, MSE is from one-step forecasts on \textbf{training set}, and \( T \) is the length of the series.

Minimizing the Gaussian AIC is asymptotically equivalent (as \( T \to \infty \)) to minimizing MSE from one-step forecasts on \textbf{test set} via time series cross-validation.
Akaike’s Information Criterion

\[ \text{AIC} = -2 \log(L) + 2k \]

Corrected AIC

For small \( T \), AIC tends to over-fit. Bias-corrected version:

\[ \text{AIC}_C = \text{AIC} + \frac{2(k+1)(k+2)}{T-k} \]

Bayesian Information Criterion

\[ \text{BIC} = \text{AIC} + k[\log(T) - 2] \]

- BIC penalizes terms more heavily than AIC
- Minimizing BIC is consistent if there is a true model.
Akaike’s Information Criterion

\[ \text{AIC} = -2 \log(L) + 2k \]

Corrected AIC
For small \( T \), AIC tends to over-fit. Bias-corrected version:

\[ \text{AIC}_C = \text{AIC} + \frac{2(k+1)(k+2)}{T-k} \]

Bayesian Information Criterion

\[ \text{BIC} = \text{AIC} + k[\log(T) - 2] \]

- BIC penalizes terms more heavily than AIC
- Minimizing BIC is consistent if there is a true model.
Akaike’s Information Criterion

**AIC**

\[ \text{AIC} = -2 \log(L) + 2k \]

**Corrected AIC**

For small \( T \), AIC tends to over-fit. Bias-corrected version:

\[ \text{AIC}_C = \text{AIC} + \frac{2(k+1)(k+2)}{T-k} \]

**Bayesian Information Criterion**

\[ \text{BIC} = \text{AIC} + k[\log(T) - 2] \]

- BIC penalizes terms more heavily than AIC
- Minimizing BIC is consistent if there is a true model.
What to use?

Choice: AIC, AICc, BIC, CV-MSE

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large $T$.
- As $T \to \infty$, BIC selects true model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.

Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.
What to use?

**Choice: AIC, AICc, BIC, CV-MSE**

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large $T$.
- As $T \to \infty$, BIC selects *true* model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.
- Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.
What to use?

**Choice: AIC, AICc, BIC, CV-MSE**

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large $T$.
- As $T \to \infty$, BIC selects *true* model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.
- Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.
What to use?

Choice: AIC, AICc, BIC, CV-MSE

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large $T$.
- As $T \to \infty$, BIC selects *true* model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.

- Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.
What to use?

Choice: AIC, AICc, BIC, CV-MSE

- CV-MSE too time consuming for most automatic forecasting purposes. Also requires large $T$.
- As $T \to \infty$, BIC selects \textit{true} model if there is one. But that is never true!
- AICc focuses on forecasting performance, can be used on small samples and is very fast to compute.
- Empirical studies in forecasting show AIC is better than BIC for forecast accuracy.
ets algorithm in R

Based on Hyndman, Koehler, Snyder & Grose (IJF 2002):

- Apply each of 15 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.
ets algorithm in R

Based on Hyndman, Koehler, Snyder & Grose (IJF 2002):

- Apply each of 15 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.
Based on Hyndman, Koehler, Snyder & Grose (IJF 2002):

- Apply each of 15 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.
Based on Hyndman, Koehler, Snyder & Grose (IJF 2002):

- Apply each of 15 models that are appropriate to the data. Optimize parameters and initial values using MLE.
- Select best method using AICc.
- Produce forecasts using best method.
- Obtain prediction intervals using underlying state space model.
Exponential smoothing

Forecasts from ETS(M,A,N)

Year
millions of sheep
300 400 500 600

Automatic algorithms for time series forecasting
Exponential smoothing
Exponential smoothing

fit <- ets(livestock)
fcast <- forecast(fit)
plot(fcast)
Exponential smoothing

Forecasts from ETS(M,N,M)

<table>
<thead>
<tr>
<th>Year</th>
<th>Total scripts (millions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>0.4</td>
</tr>
<tr>
<td>2000</td>
<td>0.6</td>
</tr>
<tr>
<td>2005</td>
<td>0.8</td>
</tr>
<tr>
<td>2010</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Exponential smoothing

Automatic algorithms for time series forecasting

Exponential smoothing

20
Exponential smoothing

fit <- ets(h02)
fcast <- forecast(fit)
plot(fcast)
> fit
ETS(M,N,M)

Smoothing parameters:
alpha = 0.4597
gamma = 1e-04

Initial states:
l = 0.4501
s = 0.8628 0.8193 0.7648 0.7675 0.6946 1.2921
   1.3327 1.1833 1.1617 1.0899 1.0377 0.9937

sigma: 0.0675

AIC     AICc     BIC
-115.69960 -113.47738 -69.24592
## M3 comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>MAPE</th>
<th>sMAPE</th>
<th>MASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theta</td>
<td>17.42</td>
<td>12.76</td>
<td>1.39</td>
</tr>
<tr>
<td>ForecastPro</td>
<td>18.00</td>
<td>13.06</td>
<td>1.47</td>
</tr>
<tr>
<td>ForecastX</td>
<td>17.35</td>
<td>13.09</td>
<td>1.42</td>
</tr>
<tr>
<td>Automatic ANN</td>
<td>17.18</td>
<td>13.98</td>
<td>1.53</td>
</tr>
<tr>
<td>B-J automatic</td>
<td>19.13</td>
<td>13.72</td>
<td>1.54</td>
</tr>
<tr>
<td><strong>ETS</strong></td>
<td>17.38</td>
<td>13.13</td>
<td>1.43</td>
</tr>
</tbody>
</table>
7 Exponential smoothing

Exponential smoothing was proposed in the late 1950s (Brown 1959, Holt 1957 and Winters 1960 are key pioneering works) and has motivated some of the most successful forecasting methods. Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older. In other words, the more recent the observation the higher the associated weight. This framework generates reliable forecasts quickly and for a wide spectrum of time series which is a great advantage and of major importance to applications in industry.

This chapter is divided into two parts. In the first part we present in detail the mechanics of all exponential smoothing methods and their application in forecasting time series with various characteristics. This is key in understanding the intuition behind these methods. In this setting, selecting and using a forecasting method may appear to be somewhat ad-hoc. The
7 Exponential smoothing

Exponential smoothing was proposed in the late 1950s (Brown 1959, Holt 1957 and Winters 1960 are key pioneering works) and has motivated some of the most successful forecasting methods. Forecasts produced using exponential smoothing methods are subject to a form of data snooping bias, with the weights decreasing as the more recent observations are included. This framework generates reliable forecasts quickly and for a wide spectrum of time series which is a great advantage and of major importance to applications in industry.

This chapter is divided into two parts. In the first part we present in detail the mechanics of all exponential smoothing methods and their application in forecasting time series with various characteristics. This is key in understanding the intuition behind these methods. In this setting, selecting and using a forecasting method may appear to be somewhat ad-hoc. The www.OTexts.org/fpp

Feedback on this book
Rob J Hyndman
George Athanasopoulos

Forecasting: principles and practice
7 Exponential smoothing

Exponential smoothing was proposed in the late 1950s (Brown 1959 and Winters 1960 are key pioneering works) and has motivated a large number of subsequent developments. It is one of the most successful forecasting methods. Forecasts produced using exponential smoothing methods are weighted averages of past observations, with observation weights decaying exponentially as the observations get older. In other words, the more recent the observation the higher the associated weight. The framework generates reliable forecasts quickly and for a wide spectrum of time series which is a great advantage and of major importance in industry.

This chapter is divided into two parts. In the first part we present the mechanics of all exponential smoothing methods and their application to forecasting time series with various characteristics. This is key in developing an understanding of the intuition behind these methods. In this setting, an understanding and using a forecasting method may appear to be somewhat ad-hoc.

1 Motivation
2 Exponential smoothing
3 ARIMA modelling
4 Automatic nonlinear forecasting?
5 Time series with complex seasonality
6 Hierarchical and grouped time series
7 The future of forecasting
ARIMA models

Inputs

\[ y_{t-1}, y_{t-2}, y_{t-3} \]

Output

\[ y_t \]
ARIMA models

Inputs

\[ y_{t-1}, y_{t-2}, y_{t-3}, \varepsilon_t \]

Output

\[ y_t \]

Autoregression (AR) model
ARIMA models

Inputs

\[ y_{t-1} \]
\[ y_{t-2} \]
\[ y_{t-3} \]

\[ \varepsilon_t \]
\[ \varepsilon_{t-1} \]
\[ \varepsilon_{t-2} \]

Output

\[ y_t \]

Autoregression moving average (ARMA) model
ARIMA models

**Inputs**
- $y_{t-1}$
- $y_{t-2}$
- $y_{t-3}$
- $\varepsilon_t$
- $\varepsilon_{t-1}$
- $\varepsilon_{t-2}$

**Output**
- $y_t$

**Estimation**
Compute likelihood $L$ from $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$.
Use optimization algorithm to maximize $L$.

**Autoregression moving average (ARMA) model**

Automatic algorithms for time series forecasting

ARIMA modelling
ARIMA models

Inputs

- $y_{t-1}$
- $y_{t-2}$
- $y_{t-3}$
- $\varepsilon_t$
- $\varepsilon_{t-1}$
- $\varepsilon_{t-2}$

Output

$y_t$

Autoregression moving average (ARMA) model applied to differences.

Estimation

Compute likelihood $L$ from $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$. Use optimization algorithm to maximize $L$. 

Automatic algorithms for time series forecasting
Automatic ARIMA modeling including interventions, using time series expert software

G. Mélard*, J.-M. Pasteels

ISRO CP 210 (bldg NO room 2.O.9.300), Campus Plaine, Université Libre de Bruxelles, Bd du Triomphe, B-1050 Bruxelles, Belgium

Abstract

This article has three objectives: (a) to describe the method of automatic ARIMA modeling (AAM), with and without intervention analysis, that has been used in the analysis; (b) to comment on the results; and (c) to comment on the M3 Competition in general. Starting with a computer program for fitting an ARIMA model and a methodology for building univariate ARIMA models, an expert system has been built, while trying to avoid the pitfalls of most existing software packages. A software package called Time Series Expert TSE-AX is used to build a univariate ARIMA model with or without an intervention analysis. The characteristics of TSE-AX are summarized and, more especially, its automatic ARIMA modeling method. The motivation to take part in the M3-Competition is also outlined. The methodology is described mainly...
Automatic Modeling Methods
for Univariate Series

Víctor Gómez
Ministerio de Hacienda

Agustín Maravall
Banco de España
Automatic Time Series Forecasting: The forecast Package for R

Rob J. Hyndman
Monash University

Yeasmin Khandakar
Monash University

Abstract

Automatic forecasts of large numbers of univariate time series are often needed in business and government. We describe an automated forecasting algorithm to do this in R, a free and open-source software environment.
Forecasts from ARIMA(0,1,0) with drift

Year
millions of sheep
250 300 350 400 450 500 550
Auto ARIMA

Forecasts from ARIMA(0,1,0) with drift

```
fit <- auto.arima(livestock)
fcast <- forecast(fit)
plot(fcast)
```
Auto ARIMA

Forecasts from ARIMA(3,1,3)(0,1,1)[12]

Year
Total scripts (millions)
1995 2000 2005 2010
0.4 0.6 0.8 1.0 1.2 1.4
```r
fit <- auto.arima(h02)
fcast <- forecast(fit)
plot(fcast)
```
> fit
Series: h02
ARIMA(3,1,3)(0,1,1)[12]

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>ar2</th>
<th>ar3</th>
<th>ma1</th>
<th>ma2</th>
<th>ma3</th>
<th>sma1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.3648</td>
<td>0.0636</td>
<td>0.3568</td>
<td>-0.4850</td>
<td>0.0479</td>
<td>-0.353</td>
<td>-0.5931</td>
</tr>
<tr>
<td>s.e.</td>
<td>0.2198</td>
<td>0.3293</td>
<td>0.1268</td>
<td>0.2227</td>
<td>0.2755</td>
<td>0.212</td>
<td>0.0651</td>
</tr>
</tbody>
</table>

sigma^2 estimated as 0.002706: log likelihood=290.25
AIC=-564.5  AICc=-563.71  BIC=-538.48
How does auto.arima() work?

A non-seasonal ARIMA process

\[
\phi(B)(1 - B)^d y_t = c + \theta(B) \varepsilon_t
\]

Need to select appropriate orders \( p, q, d \), and whether to include \( c \).

Algorithm choices driven by forecast accuracy.
How does auto.arima() work?

A non-seasonal ARIMA process

\[ \phi(B)(1 - B)^d y_t = c + \theta(B) \varepsilon_t \]

Need to select appropriate orders \( p, q, d, \) and whether to include \( c. \)

Hyndman & Khandakar (JSS, 2008) algorithm:

- Select no. differences \( d \) via KPSS unit root test.
- Select \( p, q, c \) by minimising AICc.
- Use stepwise search to traverse model space, starting with a simple model and considering nearby variants.

Algorithm choices driven by forecast accuracy.
How does auto.arima() work?

A non-seasonal ARIMA process

\[ \phi(B)(1 - B)^d y_t = c + \theta(B)\varepsilon_t \]

Need to select appropriate orders \( p, q, d \), and whether to include \( c \).

Hyndman & Khandakar (JSS, 2008) algorithm:

- Select no. differences \( d \) via KPSS unit root test.
- Select \( p, q, c \) by minimising AICc.
- Use stepwise search to traverse model space, starting with a simple model and considering nearby variants.

Algorithm choices driven by forecast accuracy.
How does auto.arima() work?

A seasonal ARIMA process

\[
\Phi(B^m)\phi(B)(1 - B)^d(1 - B^m)^D y_t = c + \Theta(B^m)\theta(B)\epsilon_t
\]

Need to select appropriate orders \(p, q, d, P, Q, D\), and whether to include \(c\).

Hyndman & Khandakar (JSS, 2008) algorithm:

- Select no. differences \(d\) via KPSS unit root test.
- Select \(D\) using OCSB unit root test.
- Select \(p, q, P, Q, c\) by minimising AICc.
- Use stepwise search to traverse model space, starting with a simple model and considering nearby variants.
### M3 comparisons

<table>
<thead>
<tr>
<th>Method</th>
<th>MAPE</th>
<th>sMAPE</th>
<th>MASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theta</td>
<td>17.42</td>
<td>12.76</td>
<td>1.39</td>
</tr>
<tr>
<td>ForecastPro</td>
<td>18.00</td>
<td>13.06</td>
<td>1.47</td>
</tr>
<tr>
<td>B-J automatic</td>
<td>19.13</td>
<td>13.72</td>
<td>1.54</td>
</tr>
<tr>
<td>ETS</td>
<td>17.38</td>
<td>13.13</td>
<td>1.43</td>
</tr>
<tr>
<td>AutoARIMA</td>
<td>19.12</td>
<td>13.85</td>
<td>1.47</td>
</tr>
</tbody>
</table>
Outline

1. Motivation
2. Exponential smoothing
3. ARIMA modelling
4. Automatic nonlinear forecasting?
5. Time series with complex seasonality
6. Hierarchical and grouped time series
7. The future of forecasting
Automatic algorithms for time series forecasting

Automatic nonlinear forecasting

- Automatic ANN in M3 competition did poorly.
- Linear methods did best in the NN3 competition!
- Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.
- Some good recent work by Kourentzes and Crone on automated ANN for time series.
- Watch this space!
Automatic nonlinear forecasting

- Automatic ANN in M3 competition did poorly.
- Linear methods did best in the NN3 competition!
- Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.
- Some good recent work by Kourentzes and Crone on automated ANN for time series.
- Watch this space!
Automatic nonlinear forecasting

- Automatic ANN in M3 competition did poorly.
- Linear methods did best in the NN3 competition!
- Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.
- Some good recent work by Kourentzes and Crone on automated ANN for time series.
- Watch this space!
Automatic ANN in M3 competition did poorly.

Linear methods did best in the NN3 competition!

Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.

Some good recent work by Kourentzes and Crone on automated ANN for time series.

Watch this space!
Automatic ANN in M3 competition did poorly.

Linear methods did best in the NN3 competition!

Very few machine learning methods get published in the IJF because authors cannot demonstrate their methods give better forecasts than linear benchmark methods, even on supposedly nonlinear data.

Some good recent work by Kourentzes and Crone on automated ANN for time series.

Watch this space!
Outline

1. Motivation
2. Exponential smoothing
3. ARIMA modelling
4. Automatic nonlinear forecasting?
5. Time series with complex seasonality
6. Hierarchical and grouped time series
7. The future of forecasting
Examples

US finished motor gasoline products

Weeks

Thousands of barrels per day


6500 7000 7500 8000 8500 9000 9500
Examples

Number of calls to large American bank (7am–9pm)

Number of call arrivals

100 200 300 400

3 March 17 March 31 March 14 April 28 April 12 May

5 minute intervals
Examples

Automatic algorithms for time series forecasting

Time series with complex seasonality

Turkish electricity demand

Electricity demand (GW)

Days


10 15 20 25
TBATS model

TBATS

Trigonometric terms for seasonality
Box-Cox transformations for heterogeneity
ARMA errors for short-term dynamics
Trend (possibly damped)
Seasonal (including multiple and non-integer periods)

**TBATS model**

$y_t =$ observation at time $t$

$y_t^{(\omega)} = \begin{cases} (y_t^{\omega} - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases}$

$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t$

$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$

$b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t$

$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t$

$s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$

$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{* (i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t$

$s_{j,t}^{(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{* (i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t$
TBATS model

\[ y_t = \text{observation at time } t \]

\[ y_t^{(\omega)} = \begin{cases} 
(y_t^{\omega} - 1)/\omega & \text{if } \omega \neq 0; \\
\log y_t & \text{if } \omega = 0.
\end{cases} \]

\[ y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t \]

\[ \ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t \]

\[ b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t \]

\[ d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \]

\[ s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t} \]

\[ s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{* (i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \]

\[ s_{j,t}^{(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{* (i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \]
TBATS model

\[ y_t = \text{observation at time } t \]

\[ y_t^{(\omega)} = \begin{cases} 
(y_t^{\omega} - 1)/\omega & \text{if } \omega \neq 0; \\
\log y_t & \text{if } \omega = 0. 
\end{cases} \]

\[ y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t \]

\[ \ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t \]

\[ b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t \]

\[ d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \]

\[ s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)} \]

\[ s_{j,t}^{(i)} = \begin{cases} 
 s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \\
 -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t 
\end{cases} \]
**TBATS model**

- **Observation at time t**
  \[ y_t = \text{observation at time } t \]

- **Box-Cox transformation**
  \[ y_t^{(\omega)} = \begin{cases} 
  \frac{(y_t^{\omega} - 1)}{\omega} & \text{if } \omega \neq 0; \\
  \log y_t & \text{if } \omega = 0.
  \end{cases} \]

- **Box-Cox transformation**
  \[ y_t^{(\omega)} = \ell_t + \phi b_t + \sum_{i=1}^{M} s_{t-m_i} + d_t \]

- **M seasonal periods**
  \[ \ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t \]

- **Global and local trend**
  \[ b_t = (1 - \phi) b + \phi b_{t-1} + \beta d_t \]

- **Local trend**
  \[ d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \]

- **Box-Cox transformation**
  \[ s_{j,t} = s_{j,t-1} \cos \lambda_j + s_{j,t-1}^* \sin \lambda_j + \gamma_1 d_t \]

- **Box-Cox transformation**
  \[ s_{j,t} = -s_{j,t-1}^* \sin \lambda_j + s_{j,t-1}^* \cos \lambda_j + \gamma_2 d_t \]
TBATS model

\[ y_t = \text{observation at time } t \]

\[ y_t^{(\omega)} = \begin{cases} (y_t^\omega - 1)/\omega & \text{if } \omega \neq 0; \\ \log y_t & \text{if } \omega = 0. \end{cases} \]

\[ y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t \]

\[ \ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t \]

\[ b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t \]

\[ d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \]

\[ s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)} \]

\[ s_{j,t} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{* (i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \]

\[ s_{j,t}^{(i)} = -s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-1}^{* (i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \]
TBATS model

\[ y_t = \text{observation at time } t \]

\[ y_t^{(ω)} = \begin{cases} 
(y_t^{ω} - 1)/ω & \text{if } ω \neq 0; \\
\log y_t & \text{if } ω = 0.
\end{cases} \]

\[ y_t^{(ω)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{M} s_{t-m_i}^{(i)} + d_t \]

\[ \ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t \]

\[ b_t = (1 - \phi)b + \phi b_{t-1} + \beta d_t \]

\[ d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \]

\[ s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)} \]

\[ s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \sin \lambda_{j}^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_{j}^{(i)} + \gamma_{2}^{(i)} d_t \]

Box-Cox transformation

\( M \) seasonal periods

global and local trend

ARMA error

Fourier-like seasonal terms

Automatic algorithms for time series forecasting

Time series with complex seasonality
**TBATS model**

\[ y_t = \text{observation at time } t \]

\[ y_t^{(\omega)} = \begin{cases} 
(y_t^{\omega} - 1)/\omega & \text{if } \omega \neq 0; \\
\log y_t & \text{if } \omega = 0.
\end{cases} \]

- **Box-Cox transformation**
- **Trigonometric**
- **Box-Cox**
- **ARMA**
- **Trend**
- **Seasonal**
- **Fourier-like seasonal terms**

\[ \ell_t = \ell_{t-1} \]

\[ b_t = (1 - \phi)b_{t-1} + \epsilon_t \]

\[ d_t = \sum_{i=1}^{p} \phi_i d_{t-i} \]

\[ s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)} \]

\[ s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \sin \lambda_{j}^{(i)} + s_{j,t-1}^{(i)} \cos \lambda_{j}^{(i)} + \gamma_{2,t}^{(i)} \]

**Automatic algorithms for time series forecasting**

**Time series with complex seasonality**
Examples

```r
fit <- tbats(gasoline)
fcast <- forecast(fit)
plot(fcast)
```
Examples

```r
fit <- tbats(callcentre)
fcast <- forecast(fit)
plot(fcast)
```

Forecasts from TBATS(1, {3,1}, 0.987, {<169,5>, <845,3>})

5 minute intervals
Number of call arrivals

3 March 17 March 31 March 14 April 28 April 12 May 26 May 9 June
Examples

fit <- tbats(turk)
fcast <- forecast(fit)
plot(fcast)
Outline

1. Motivation
2. Exponential smoothing
3. ARIMA modelling
4. Automatic nonlinear forecasting?
5. Time series with complex seasonality
6. Hierarchical and grouped time series
7. The future of forecasting
A **hierarchical time series** is a collection of several time series that are linked together in a hierarchical structure.

**Examples**
- Net labour turnover
- Tourism by state and region
Hierarchical time series

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

**Examples**

- Net labour turnover
- Tourism by state and region
Hierarchical time series

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

**Examples**

- Net labour turnover
- Tourism by state and region
Hierarchical time series

- $Y_t$: 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 $t$. 

Diagram: Hierarchical structure with "Total", "A", "B", and "C".
Hierarchical time series

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

$y_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]'$

$Y_t$: 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 $t$.

$$
\begin{pmatrix}
1 & 1 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
Y_{A,t} \\
Y_{B,t} \\
Y_{C,t}
\end{pmatrix}
$$
Hierarchical time series

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

- \( Y_t \): 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 time series

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

- \( Y_t \) : 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 \( t \).

\( y_t \) : observed aggregate of all series at time \( t \).
Hierarchical time series

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

\[ \mathbf{y}_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}]' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \end{pmatrix} = \mathbf{S} b_t \]
Hierarchical time series

\[ \mathbf{y}_t = \begin{pmatrix} Y_t \\ Y_{A,t} \\ Y_{B,t} \\ Y_{C,t} \\ Y_{AX,t} \\ Y_{AY,t} \\ Y_{AZ,t} \\ Y_{BX,t} \\ Y_{BY,t} \\ Y_{BZ,t} \\ Y_{CX,t} \\ Y_{CY,t} \\ Y_{CZ,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} b_t \\ s \end{pmatrix} \]
Hierarchical time series

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

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

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

\[ y_t = S b_t \]
Forecasting notation

Let \( \hat{y}_n(h) \) be vector of initial \( h \)-step forecasts, made at time \( n \), stacked in same order as \( y_t \). (They may not add up.)

Reconciled forecasts are of the form:

\[
\tilde{y}_n(h) = S P \hat{y}_n(h)
\]

for some matrix \( P \).

\( P \) extracts and combines base forecasts \( \hat{y}_n(h) \) to get bottom-level forecasts.
Forecasting notation

Let $\hat{y}_n(h)$ be vector of initial $h$-step forecasts, made at time $n$, stacked in same order as $y_t$. (They may not add up.)

Reconciled forecasts are of the form:

$$\tilde{y}_n(h) = SP\hat{y}_n(h)$$

for some matrix $P$.

- $P$ extracts and combines base forecasts $\hat{y}_n(h)$ to get bottom-level forecasts.
- $S$ adds them up.
Forecasting notation

Let $\hat{y}_n(h)$ be vector of initial $h$-step forecasts, made at time $n$, stacked in same order as $y_t$. (They may not add up.)

Reconciled forecasts are of the form:

$$\tilde{y}_n(h) = SP\hat{y}_n(h)$$

for some matrix $P$. 

- $P$ extracts and combines base forecasts $\hat{y}_n(h)$ to get bottom-level forecasts.
- $S$ adds them up.
Forecasting notation

Let \( \hat{y}_n(h) \) be vector of initial \( h \)-step forecasts, made at time \( n \), stacked in same order as \( y_t \). (They may not add up.)

Reconciled forecasts are of the form:

\[
\tilde{y}_n(h) = SP\hat{y}_n(h)
\]

for some matrix \( P \).

- \( P \) extracts and combines base forecasts \( \hat{y}_n(h) \) to get bottom-level forecasts.
- \( S \) adds them up.
Forecasting notation

Let $\hat{y}_n(h)$ be vector of initial $h$-step forecasts, made at time $n$, stacked in same order as $y_t$. (They may not add up.)

Reconciled forecasts are of the form:

$$\tilde{y}_n(h) = SP\hat{y}_n(h)$$

for some matrix $P$.

- $P$ extracts and combines base forecasts $\hat{y}_n(h)$ to get bottom-level forecasts.
- $S$ adds them up
General properties

\[ \tilde{y}_n(h) = SP\hat{y}_n(h) \]

Forecast bias
Assuming the base forecasts \( \hat{y}_n(h) \) are unbiased, then the revised forecasts are unbiased iff \( SPS = S \).

Forecast variance
For any given \( P \) satisfying \( SPS = S \), the covariance matrix of the \( h \)-step ahead reconciled forecast errors is given by

\[ \text{Var}[y_{n+h} - \tilde{y}_n(h)] = SPW_h P' S' \]

where \( W_h \) is the covariance matrix of the \( h \)-step ahead base forecast errors.
General properties

\[ \hat{y}_n(h) = SP\hat{y}_n(h) \]

Forecast bias

Assuming the base forecasts \( \hat{y}_n(h) \) are unbiased, then the revised forecasts are unbiased iff \( SPS = S \).

Forecast variance

For any given \( P \) satisfying \( SPS = S \), the covariance matrix of the \( h \)-step ahead reconciled forecast errors is given by

\[ \text{Var}[y_{n+h} - \hat{y}_n(h)] = SPW_hP'S' \]

where \( W_h \) is the covariance matrix of the \( h \)-step ahead base forecast errors.
General properties

\[ \tilde{y}_n(h) = SP\hat{y}_n(h) \]

**Forecast bias**

Assuming the base forecasts \( \hat{y}_n(h) \) are unbiased, then the revised forecasts are unbiased iff \( SPS = S \).

**Forecast variance**

For any given \( P \) satisfying \( SPS = S \), the covariance matrix of the \( h \)-step ahead reconciled forecast errors is given by

\[ \text{Var}[y_{n+h} - \tilde{y}_n(h)] = SPW_hP'S' \]

where \( W_h \) is the covariance matrix of the \( h \)-step ahead base forecast errors.
Theorem
For any $P$ satisfying $SPS = S$, then
$$\min_P \text{trace}[SPW_h P'S']$$
has solution $P = \left(S'W_h^\dagger S\right)^{-1}S'W_h^\dagger$.

$W_h^\dagger$ is generalized inverse of $W_h$.

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

Revised forecasts Base forecasts
BLUF via trace minimization

Theorem

For any \( P \) satisfying \( SPS = S \), then

\[
\min_P = \text{trace}[SPW_hP'S']
\]

has solution \( P = (S'W_hS)^{-1}S'W_h \).

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

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

Revised forecasts

Base forecasts

Equivalent to GLS estimate of regression:

\[
\hat{y}_n(h) = S\beta_n(h) + \epsilon_n \text{ where } \epsilon_n \sim N(0, W_h).
\]
Theorem

For any $P$ satisfying $SPS = S$, then

$$\min_P = \text{trace}[SPW_h P'S']$$

has solution $P = (S'W_h^+ S)^{-1} S'W_h^+$. 

- $W_h^+$ is generalized inverse of $W_h$.

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

Revised forecasts

- Equivalent to GLS estimate of regression
  \[
  \hat{y}_n(h) = S\beta_n(h) + \epsilon_h \text{ where } \epsilon \sim N(0, W_h).
  \]

Base forecasts

- Problem: $W_h^+$ hard to estimate.
BLUF via trace minimization

**Theorem**

For any $P$ satisfying $SPS = S$, then

$$\min_P = \text{trace}[SPW_hP'S']$$

has solution $P = (S'W_h^†S)^{-1}S'W_h^†$.

- $W_h^†$ is generalized inverse of $W_h$.

$$\hat{y}_n(h) = S(S'W_h^†S)^{-1}S'W_h^†\hat{y}_n(h)$$

**Revised forecasts**

- Equivalent to GLS estimate of regression
  $$\hat{y}_n(h) = S\beta_n(h) + \varepsilon_h \text{ where } \varepsilon \sim N(0, W_h).$$

**Problem:** $W_h$ hard to estimate.

Automatic algorithms for time series forecasting  
Hierarchical and grouped time series  
51
Theorem

For any $P$ satisfying $SPS = S$, then

$$\min_P \text{trace}[SPW_h P' S']$$

has solution $P = (S'W_h S)^{-1} S' W_h$.

- $W_h^\dagger$ is generalized inverse of $W_h$.

- Revised forecasts $\tilde{y}_n(h) = S (S'W_h S)^{-1} S'W_h \hat{y}_n(h)$
  - Equivalent to GLS estimate of regression $\hat{y}_n(h) = S\beta_n(h) + \varepsilon_h$ where $\varepsilon \sim N(0, W_h)$.
  - Problem: $W_h$ hard to estimate.
Theorem
For any $P$ satisfying $SPS = S$, then

$$\min_P = \text{trace}[SPW_P P'S']$$

has solution $P = (S'W_P S)^{-1}S'W_P$.

- $W_P^\dagger$ is generalized inverse of $W_P$.

Revised forecasts
- Equivalent to GLS estimate of regression
  $\hat{y}_n(h) = S\beta_n(h) + \epsilon_h$ where $\epsilon \sim N(0, W_h)$.

Problem: $W_h$ hard to estimate.
Optimal combination forecasts

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

Revised forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate \( W_1 \) by its diagonal.
- Assume \( W_h = k_h W_1 \).
- Easy to estimate, and places weight where we have best one-step forecasts:

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

Revised forecasts \( \tilde{y}_n(h) = S(S'W_h^SW)^{-1}S'W_h\hat{y}_n(h) \)

Base forecasts

**Solution 1: OLS**

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

**Solution 2: WLS**

- Approximate \( W_1 \) by its diagonal.
- Assume \( W_h = k_hW_1 \).
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Revised forecasts

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

Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate \( W_1 \) by its diagonal.
- Assume \( W_h = k_h W_1 \).
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Revised forecasts

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

Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate \( W_1 \) by its diagonal.
- Assume \( W_h = k_h W_1 \).
- Easy to estimate, and places weight where we have best one-step forecasts.

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

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

Revised forecasts

Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate \( W_1 \) by its diagonal.
- Assume \( W_h = k_h W_1 \).
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Solution 1: OLS

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

Solution 2: WLS

- Approximate \( W_1 \) by its diagonal.
- Assume \( W_h = k_h W_1 \).
- Easy to estimate, and places weight where we have best one-step forecasts.

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

Revised forecasts

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

Base forecasts

Solution 1: OLS

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

Solution 2: WLS

- Approximate \( W_1 \) by its diagonal.
- Assume \( W_h = k_h W_1 \).
- Easy to estimate, and places weight where we have best one-step forecasts.

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

- Computational difficulties in big hierarchies due to size of the $S$ matrix and singular behavior of $(S' \Lambda S)$.
- Loss of information in ignoring covariance matrix in computing point forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

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

- Computational difficulties in big hierarchies due to size of the $S$ matrix and singular behavior of $(S' \Lambda S)$.

- Loss of information in ignoring covariance matrix in computing point forecasts.

- Still need to estimate covariance matrix to produce prediction intervals.
Challenges

- Computational difficulties in big hierarchies due to size of the $S$ matrix and singular behavior of $(S' \Lambda S)$.
- Loss of information in ignoring covariance matrix in computing point forecasts.
- Still need to estimate covariance matrix to produce prediction intervals.

$$\tilde{y}_n(h) = S(S' \Lambda S)^{-1}S' \Lambda \hat{y}_n(h)$$
Australian tourism

Automatic algorithms for time series forecasting

Hierarchical and grouped time series

54
Australian tourism

Hierarchy:

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

< -3%
-3% to 0
0 to 3%
> 3%
Australian tourism

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

Base forecasts
ETS (exponential smoothing) models

0 to 3%
> 3%
Base forecasts

Domestic tourism forecasts: Total

Visitor nights

Year


60000 65000 70000 75000 80000 85000

Automatic algorithms for time series forecasting Hierarchical and grouped time series
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>
</tbody>
</table>

Year
Visitor nights
18000 22000 26000 30000
Domestic tourism forecasts: VIC

Visitor nights

Year


10000 12000 14000 16000 18000
Base forecasts

Domestic tourism forecasts: Nth.Coast.NSW

Year
Visitor nights
1998
2000
2002
2004
2006
2008
Base forecasts

Domestic tourism forecasts: Metro.QLD

Visitor nights

Year


8000 9000 11000 13000
Domestic tourism forecasts: Sth.WA

Visitor nights

Year

400 600 800 1000 1200 1400
Base forecasts

Domestic tourism forecasts: X201.Melbourne

Visitor nights over years from 1998 to 2008.
Reconciled forecasts

![Graph showing time series data with peaks and troughs.]
Reconciled forecasts

![Graphs showing time series data for NSW, VIC, QLD, and Other categories from 2000 to 2010.](image)
Reconciled forecasts

Automatic algorithms for time series forecasting

Hierarchical and grouped time series

Sydney

Other NSW

Melbourne

Other VIC

Other QLD

Capital cities

Other

Sydney

Other NSW

Melbourne

Other VIC

Other QLD

Capital cities

Other
Select models using all observations;

- Re-estimate models using first 12 observations and generate 1- to 8-step-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-step-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-step-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-step-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

<table>
<thead>
<tr>
<th>MAPE</th>
<th>$h = 1$</th>
<th>$h = 2$</th>
<th>$h = 4$</th>
<th>$h = 6$</th>
<th>$h = 8$</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Top Level: Australia</strong></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>
</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>
</tr>
<tr>
<td>WLS</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>
</tr>
<tr>
<td><strong>Level: States</strong></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>
</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>
</tr>
<tr>
<td>WLS</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>
</tr>
<tr>
<td><strong>Level: Zones</strong></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>
</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>
</tr>
<tr>
<td>WLS</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>
</tr>
<tr>
<td><strong>Bottom Level: Regions</strong></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>
</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>
</tr>
<tr>
<td>WLS</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>
</tr>
</tbody>
</table>
hts: Hierarchical and grouped time series
Methods for analysing and forecasting hierarchical and grouped time series

Version: 4.5
Depends: forecast (≥ 5.0), SparseM
Imports: parallel, utils
Published: 2014-12-09
Author: Rob J Hyndman, Earo Wang and Alan Lee
Maintainer: Rob J Hyndman <Rob.Hyndman at monash.edu>
BugReports: https://github.com/robjhyndman/hts/issues
License: GPL (≥ 2)
Outline

1. Motivation
2. Exponential smoothing
3. ARIMA modelling
4. Automatic nonlinear forecasting?
5. Time series with complex seasonality
6. Hierarchical and grouped time series
7. The future of forecasting
Forecasts about forecasting

1. Automatic algorithms will become more general — handling a wide variety of time series.

2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.

3. Automatic forecasting algorithms for multivariate time series will be developed.

4. Automatic forecasting algorithms that include covariate information will be developed.
Forecasts about forecasting

1 Automatic algorithms will become more general — handling a wide variety of time series.

2 Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.

3 Automatic forecasting algorithms for multivariate time series will be developed.

4 Automatic forecasting algorithms that include covariate information will be developed.
Automatic algorithms will become more general — handling a wide variety of time series.

Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.

Automatic forecasting algorithms for multivariate time series will be developed.

Automatic forecasting algorithms that include covariate information will be developed.
Forecasts about forecasting

1. Automatic algorithms will become more general — handling a wide variety of time series.

2. Model selection methods will take account of multi-step forecast accuracy as well as one-step forecast accuracy.

3. Automatic forecasting algorithms for multivariate time series will be developed.

4. Automatic forecasting algorithms that include covariate information will be developed.
For further information

robjhyndman.com

- Slides and references for this talk.
- Links to all papers and books.
- Links to R packages.
- A blog about forecasting research.