Smoothing Methods for Time Series with Trend

Lesson 2.3

Introduction

Previously, we learned that both moving average and simple exponential smoothing should only be used for forecasting series with no trend or seasonality. One solution for forecasting series with trend and/or seasonality is first to remove those components (e.g., via differencing). Another solution is to use a more sophisticated version of exponential smoothing, which can capture trend and/or seasonality.

Holt's linear trend method

Holt (1957) extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations: one for the level and one for the trend

$$\begin{aligned} \text{Forecast equation} : \hat{y}_{t+h|t} &= l_t + hb_t \\ \text{Level equation} : l_t &= \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) \\ \text{Trend equation} : b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \end{aligned}$$

Here l_t denotes an estimate of the level of the series at time t, b_t denotes an estimate of the trend (slope) of the series at time t, α is the smoothing parameter for the level, $0 \leq \alpha \leq 1$, and β^* is the smoothing parameter for the trend, $0 \leq \beta^* \leq 1$. As with simple exponential smoothing, the level equation shows that l_t is a wieghted average of the observations y_t and the one-step-ahead forecast for time t which is given by $l_{t-1} + b_{t-1}$.

This method is also, known as *double exponential smoothing*. The trend equation shows that b_t is a weighted average of the estimated trend at time t based on $l_t - l_{t-1}$ and b_{t-1}, the previous estimate of trend. Here the forecast function is no longer flat but trending. In addition, the *h-step-ahead* forecast is equal to the last estimated level plus h times the last estimated trend value. Hence, the forecasts are a linear function of h

To illustrate these ideas, consider the Australian air passenger time series.

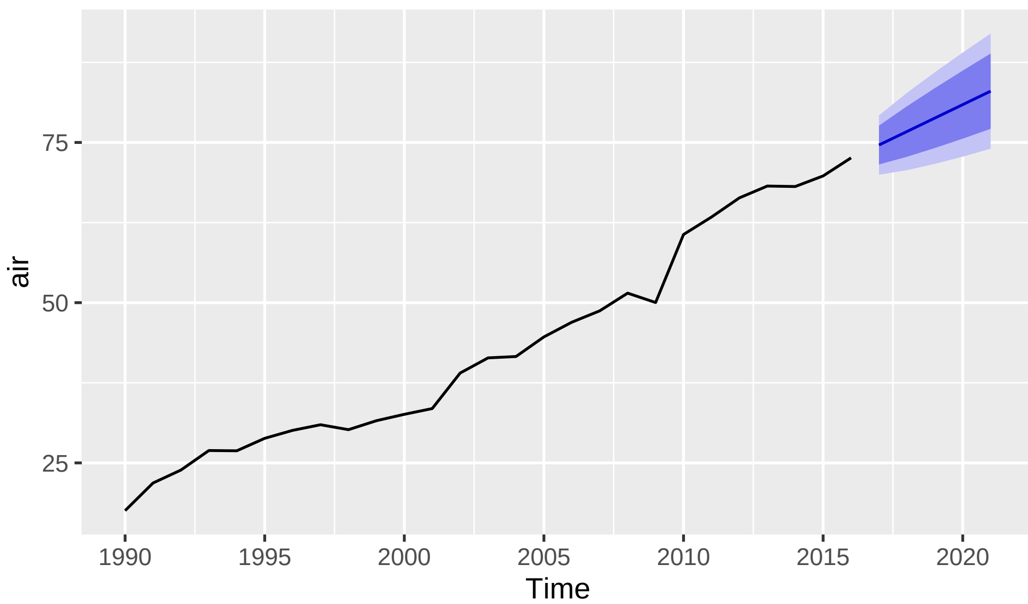
```r
library(forecast)
library(fpp2)
air <- window(ausair,start=1990)#From 1990 to 2016
fc <- holt(air,h=5)#Forecast from 2017 to 2021
fc$model#Gives the parameter estimates and the initial state (t=0) values
```

```
Holt's method

Call:
holt(y = air, h = 5)

  Smoothing parameters:
    alpha = 0.8302
    beta  = 1e-04

  Initial states:
    l = 15.5715
    b = 2.1017

  sigma:  2.3645

     AIC     AICc      BIC
141.1291 143.9863 147.6083
```

```r
print(fc)#displays the 5-year forecast
```

```
     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2017       74.60130 71.57106 77.63154 69.96695 79.23566
2018       76.70304 72.76440 80.64169 70.67941 82.72668
2019       78.80478 74.13092 83.47864 71.65673 85.95284
2020       80.90652 75.59817 86.21487 72.78810 89.02494
2021       83.00826 77.13343 88.88310 74.02348 91.99305
```

```r
autoplot(air) +
  autolayer(fc, PI=T)
```



Damped trend methods

The forecasts generated by Holt's linear method display a constant trend (increasing or decreasing) indefinitely into the future. Empirical evidence indicates that these methods tend to over-forecast, especially for longer forecast horizons. Gardner & McKenzie (1985) introduced a parameter that "dampens" the trend to a flat line some time in the future.

In conjunction with the smoothing parameters α and β^* this method also includes a damping parameter $0 < \phi < 1$.

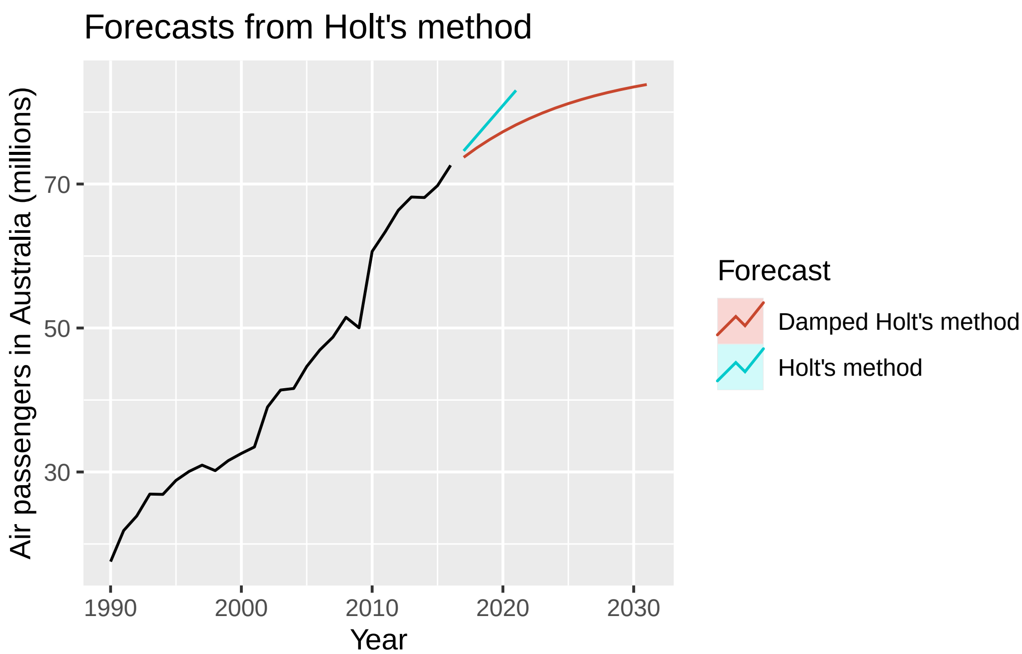
$$\begin{aligned}
\hat{y}_{t+h|t} &= l_t + (\phi + \phi^2 + \cdots + \phi^h)b_t \\
l_t &= \alpha y_t + (1-\alpha)(l_{t-1} + \phi b_{t-1}) \\
b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1}
\end{aligned}$$

If $\phi = 1$ the method is identical to Holt's linear method.

In practice, ϕ is rarely less than 0.8 as the damping has a very strong effect for smaller values. Values of *phi* close to 1 will mean that a damped model is not able to be distinguished from a non-damped model. For these reasons, we usually restrict $0.8 < \phi < 0.98$

As example, consider gain the passenger data.

```
fc1 <- holt(air, h=15)
fc2 <- holt(air, damped=TRUE, phi = 0.9, h=15)
autoplot(air) +
  autolayer(fc, series="Holt's method", PI=FALSE) +
  autolayer(fc2, series="Damped Holt's method", PI=FALSE) +
  ggtitle("Forecasts from Holt's method") + xlab("Year") +
  ylab("Air passengers in Australia (millions)") +
  guides(colour=guide_legend(title="Forecast"))
```



We compare the forecasting performance of the three exponential smoothing methods that we have considered so far in forecasting the sheep livestock population in Asia.

- We will use time series cross-validation, *tsCV()*, function in the forecast package to compare the one-step forecast accuracy of the three methods

```
e1 <- tsCV(livestock, ses, h=1)
e2 <- tsCV(livestock, holt, h=1)
e3 <- tsCV(livestock, holt, damped=TRUE, h=1)
MSE1 <- mean(e1^2, na.rm=TRUE)
MSE2 <- mean(e2^2, na.rm=TRUE)
MSE3 <- mean(e3^2, na.rm=TRUE)
MAE1 <- mean(abs(e1), na.rm=TRUE)
MAE2 <- mean(abs(e2), na.rm=TRUE)
```

```r
MAE3 <- mean(abs(e3), na.rm=TRUE)
print(cbind(MSE1,MSE2,MSE3,MAE1,MAE2,MAE3))
```

```
         MSE1    MSE2     MSE3    MAE1     MAE2     MAE3
[1,] 178.2531 173.365 162.6274 8.53246 8.803058 8.024192
```

It is shown that Damped Holt's method is best whether you compare MAE or MSE values. So we will proceed with using the damped Holt's method and apply it to the whole data set to get forecasts for future years

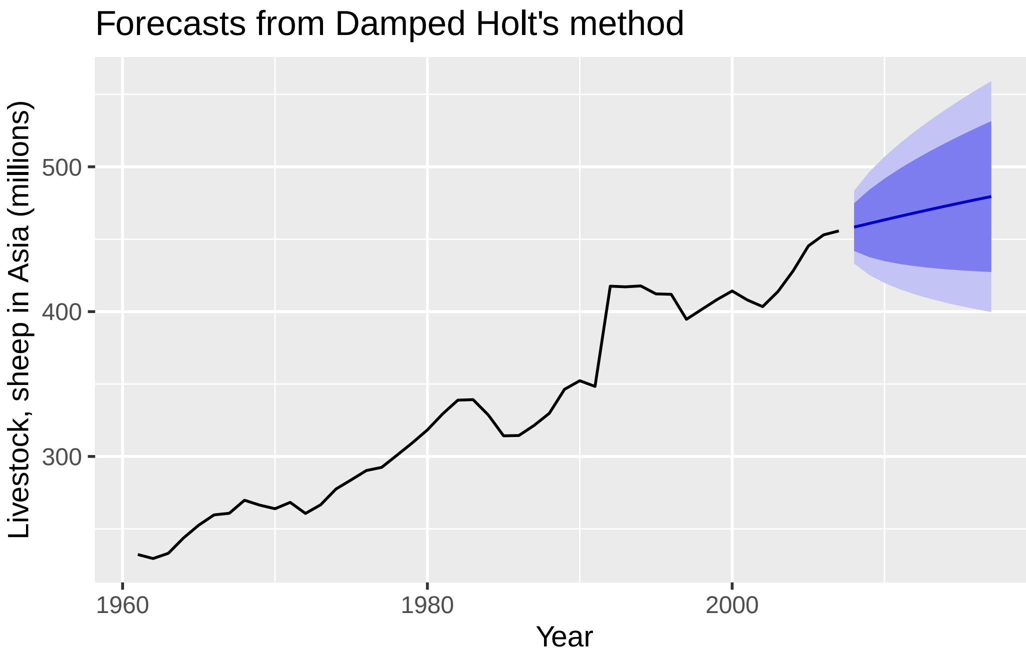
```r
fc3 <- holt(livestock, damped=TRUE)
par.fc3 <- fc3$model$par
print(par.fc3)
```

```
       alpha         beta          phi             l             b
9.998998e-01 2.806460e-04 9.797542e-01 2.233500e+02 6.904597e+00
```

```r
print(fc3)
```

```
     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2008       458.3355 441.8760 474.7951 433.1628 483.5083
2009       460.8784 437.5990 484.1578 425.2757 496.4811
2010       463.3697 434.8551 491.8844 419.7603 506.9792
2011       465.8107 432.8806 498.7407 415.4485 516.1728
2012       468.2022 431.3806 505.0237 411.8884 524.5159
2013       470.5452 430.2041 510.8864 408.8488 532.2417
2014       472.8409 429.2620 516.4198 406.1927 539.4891
2015       475.0900 428.4964 521.6837 403.8312 546.3489
2016       477.2937 427.8676 526.7198 401.7029 552.8844
2017       479.4527 427.3466 531.5588 399.7633 559.1421
```

```r
autoplot(fc3) +
  xlab("Year") + ylab("Livestock, sheep in Asia (millions)")
```


Forecasts from Damped Holt's method
Livestock, sheep in Asia (millions)
500
400
300
1960
1980
2000
Year