Session 4. Multiple Frequencies ETS

Demand Forecasting with the ADAM

Ivan Svetunkov (i.svetunkov@lancaster.ac.uk)

Contents

Introduction

Have you loaded the `greybox` and `smooth` packages?

```
library(greybox)
library(smooth)
```

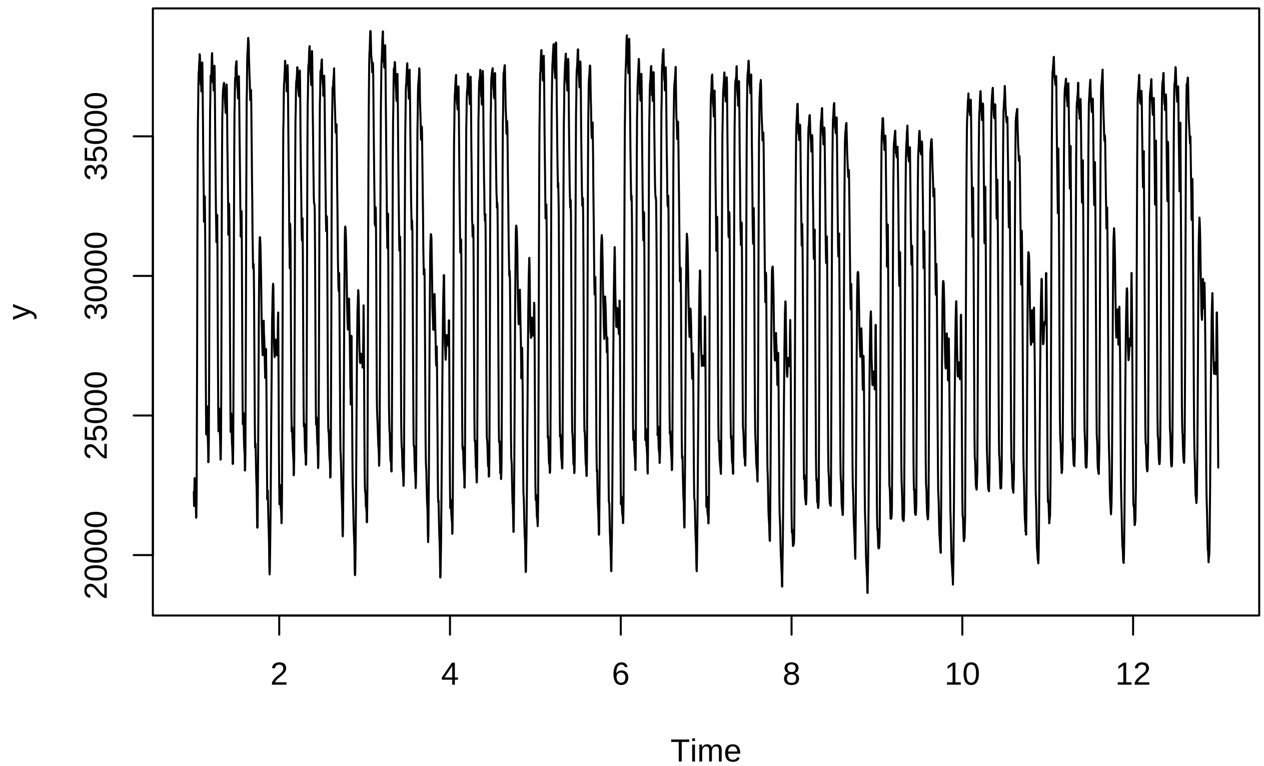
Multiple Frequencies

In this part of the workshop, we will work with half-hourly data, series `taylor` from the `forecast` package in R:

```
y <- forecast::taylor
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
plot(y)
```



The series has two seasonal patterns: half-hour of day and day of week. It does not have an apparent trend, so we will fit a double seasonal ETS(M,N,M) model. Given that we will have to estimate many seasonal indices, we will use “backcasting” as an initialisation for the model. This will speed things up. But if you have time, we encourage you to experiment with `initial="optimal"` (this is the default value):

```
adamETSMNM <- adam(y, "MNM", lags=c(48,336),
                   h=336, holdout=TRUE,
                   initial="backcasting")
adamETSMNM
```

```
## Time elapsed: 0.52 seconds
## Model estimated using adam() function: ETS(MNM)[48, 336]
## Distribution assumed in the model: Gamma
## Loss function type: likelihood; Loss function value: 25682.88
## Persistence vector g:
##  alpha gamma1 gamma2
## 0.1357 0.2813 0.2335
##
## Sample size: 3696
## Number of estimated parameters: 4
## Number of degrees of freedom: 3692
```

```
## Information criteria:
##      AIC     AICc      BIC     BICc
## 51373.76 51373.77 51398.62 51398.66
##
## Forecast errors:
## ME: 625.221; MAE: 716.941; RMSE: 817.796
## sCE: 709.966%; Asymmetry: 90.4%; sMAE: 2.423%; sMSE: 0.076%
## MASE: 1.103; RMSSE: 0.867; rMAE: 0.107; rRMSE: 0.1
```

Notice that the seasonal smoothing parameters are relatively high in this model: the second γ is equal to 0.2335, which means that the model adapts the seasonal profile to the data too often (takes 23.35% of the error from the previous observation in it). Furthermore, the smoothing parameter α equals to 0.1357, which is also potentially high, given that we have well-behaved data and that we deal with a multiplicative model. This might indicate that the model overfits the data. To see if this is the case, we can produce the plot of components over time:

```
plot(adamETSMNM, which=12)
```

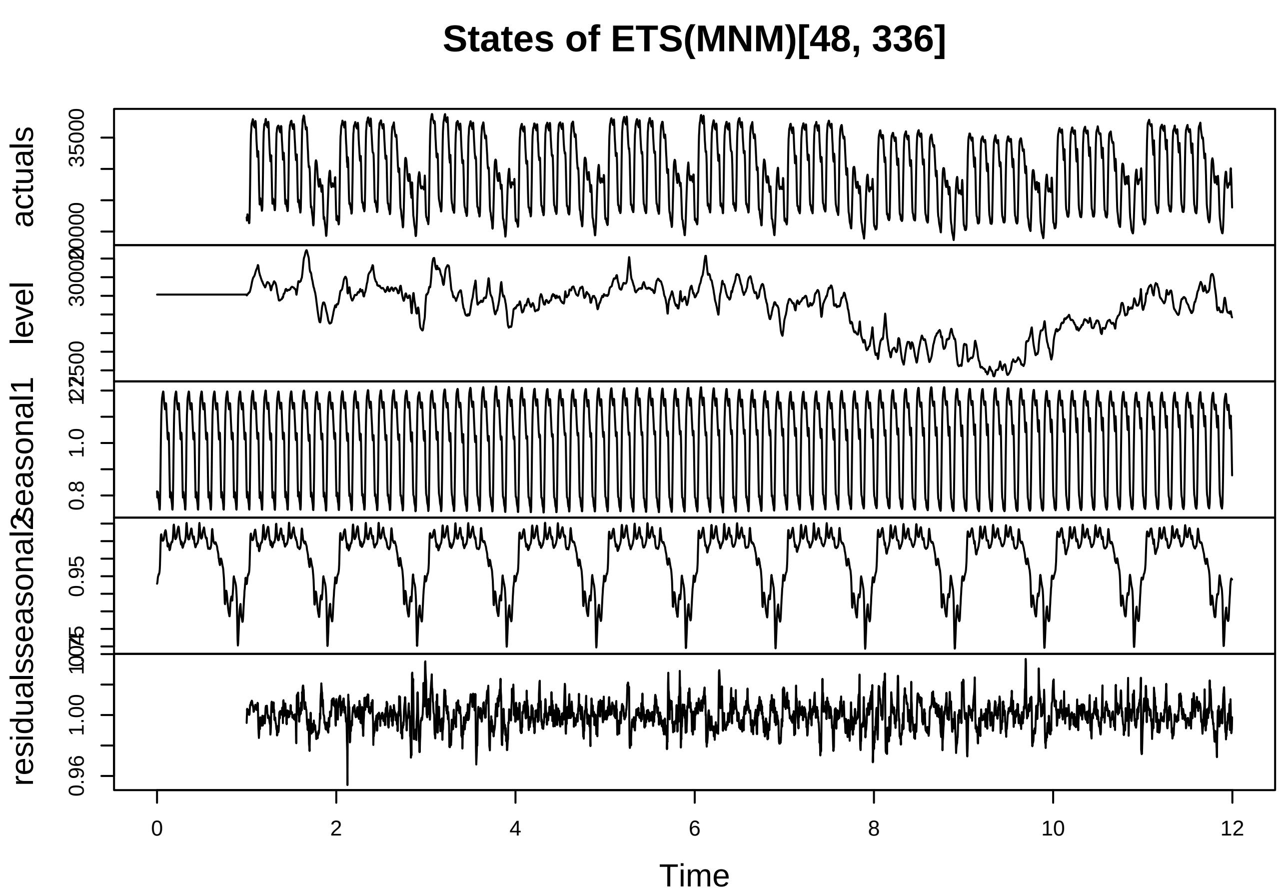


Figure 1: Half-hourly electricity demand data decomposition according to ETS(M,N,M)[48,336]

In the figure above, we see that the first seasonal component evolves over time too much, while the second contains some elements of the intra-day seasonality - in the ideal scenario the latter should not happen. Furthermore, the level seems to be repeating some parts of the seasonality. Ideally, we want to have a smooth level component and for the second seasonal component not to have those spikes for half-hour of day seasonality. In order to address some of these issues we will use a different loss function, GTMSE. Note that this time the model estimation will take much more time (it took 30 seconds on my PC):

```
adamETSMNMGTMSE <- adam(y, "MNM", lags=c(48,336),
                        h=336, holdout=TRUE,
```

```
                        initial="back", loss="GTMSE")
adamETSMNMGTMSE
```

```
## Time elapsed: 31.13 seconds
## Model estimated using adam() function: ETS(MNM)[48, 336]
## Distribution assumed in the model: Normal
## Loss function type: GTMSE; Loss function value: -2648.693
## Persistence vector g:
##  alpha gamma1 gamma2
## 0.0321 0.2602 0.1403
##
## Sample size: 3696
## Number of estimated parameters: 3
## Number of degrees of freedom: 3693
## Information criteria are unavailable for the chosen loss & distribution.
##
## Forecast errors:
## ME: 220.414; MAE: 378.701; RMSE: 507.755
## sCE: 250.29%; Asymmetry: 63.8%; sMAE: 1.28%; sMSE: 0.029%
## MASE: 0.582; RMSSE: 0.538; rMAE: 0.057; rRMSE: 0.062
```

Comparing the error measures of this model with the previous one, we can see that it does a better job in capturing the patterns in the data.

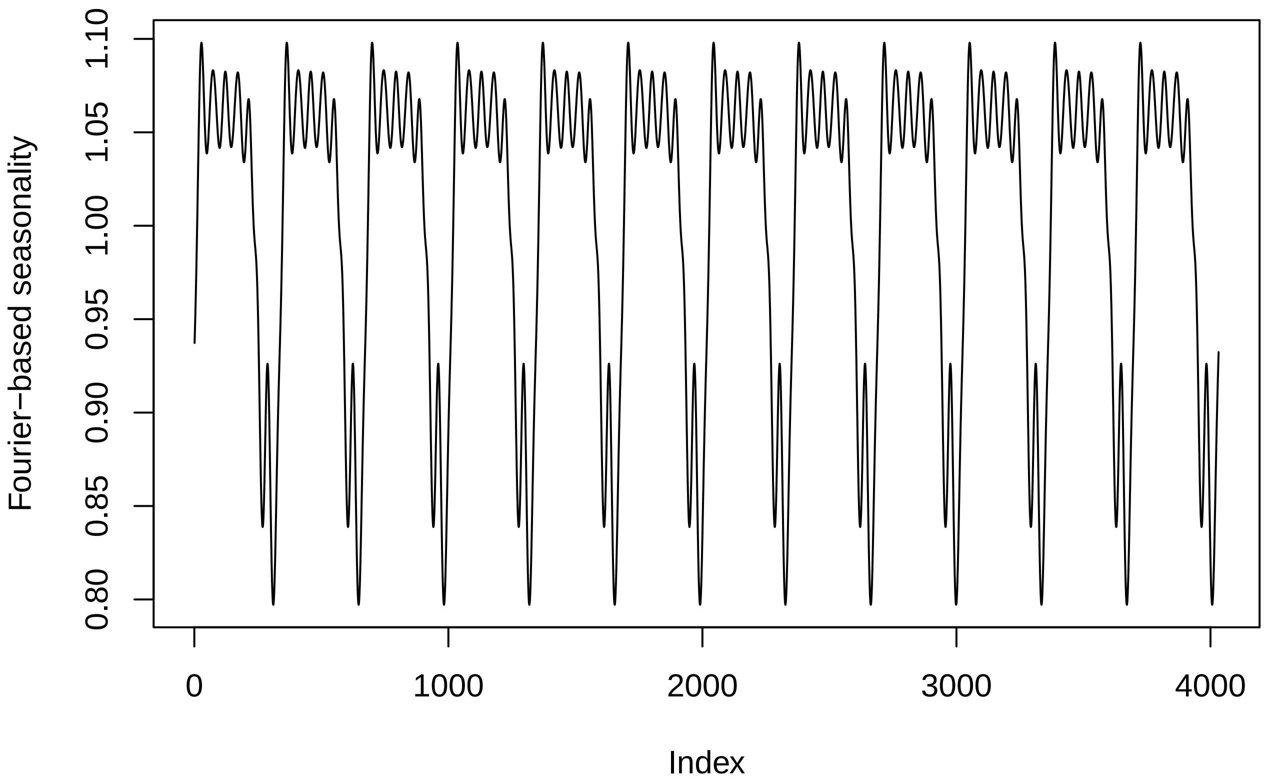
Alternatively, we can try adding Fourier terms as explanatory variables not to bother with the lag of 336 (takes approximately 10 seconds on my PC):

```
# Create fourier terms
z <- forecast::fourier(ts(y, frequency=336), K=10)
# Move them to a data.frame
yData <- data.frame(y=y, z)
# Estimate the model
adamETSMNMFourier <- adam(yData, "MNM", lags=c(48),
                        formula=y~., initial="back",
                        h=336, holdout=TRUE)
adamETSMNMFourier
```

```
## Time elapsed: 11.03 seconds
## Model estimated using adam() function: ETSX(MNM)
## Distribution assumed in the model: Gamma
## Loss function type: likelihood; Loss function value: 26278.5
## Persistence vector g (excluding xreg):
## alpha gamma
##     1     0
##
## Sample size: 3696
## Number of estimated parameters: 23
## Number of degrees of freedom: 3673
## Information criteria:
##      AIC     AICc      BIC     BICc
## 52603.00 52603.30 52745.95 52747.18
##
## Forecast errors:
## ME: 262.437; MAE: 660.143; RMSE: 927.725
## sCE: 298.009%; Asymmetry: 41.4%; sMAE: 2.231%; sMSE: 0.098%
## MASE: 1.015; RMSSE: 0.983; rMAE: 0.099; rRMSE: 0.113
```

As we see from the output above, the smoothing parameter α is too high, which leads to potential overfitting and lowers the potential accuracy of the model. The fourier terms capture the long seasonal cycle well, as can be seen from the following plot:

```
plot(exp(z %*% adamETSMNMFourier$initial$xreg),
     type="l", ylab="Fourier-based seasonality")
```



Note that the we did not allow the parameters for fourier to evolve over time. This is one of the potential limitations.

Finally, we could try TBATS to see how it compares with the models that we have applied so far. We will use `tbats()` function from the `forecast` package. Model estimation takes approximately a minute in this case, so we will not run this in the class, but you can try it at home to see how it works:

```
library(forecast)
# Get number of all observations
obs <- length(y)
# Fit the TBATS model
taylorTBATS <- msts(y[1:(obs-336)],
                    seasonal.periods=c(48,336),
                    start=start(taylor)) |>
    tbats()
# Produce forecasts
taylorTBATSForecast <- forecast(taylorTBATS, h=336, level=0.95)

# Calculate error measures, similar to how it is done in adam()
measures(y[-c(1:(obs-336))],
         taylorTBATSForecast$mean,
         y[c(1:(obs-336))]) |>
    round(5)
```

On my computer, TBATS produced a higher holdout sample RMSE than the double seasonal ADAM ETS. But this does not mean much if we only consider one origin and one time series.

Additional materials

For some additional examples on ETS implemented in `smooth` run:

```
vignette("adam","smooth")
```

Some additional resources on exponential smoothing:

1. Paper on multi-step estimators;
2. Probabilistic forecasting of hourly emergency department arrivals paper;
3. ETS in the blog of Ivan Svetunkov
4. Posts on the functions in smooth package
5. Posts of Nikos Kourentzes on exponential smoothing