ETW3420

Principles of Forecasting and Applications

Topic 5 Exercises - Part 2

Question 1

(a) Show that the forecast variance for an ETS(A,N,N) model is given by

$$\sigma^2\left[1 + \alpha^2(h-1)\right].$$

(b) Write down the corresponding 95% prediction interval as a function of ℓ_T, α, h and σ, assuming Gaussian errors.

Question 2

For this question, use the quarterly UK passenger vehicle production data from 1977Q1–2005Q1 (data set `ukcars`).

(a) Plot the data and describe the main features of the series.

(b) Use `ets()` to choose a seasonal model for the data.

(c) Check the residuals of the ETS model.

(d) Produce and plot the forecasts for h = 24 from the fitted ets model. Comment on why the forecasts show no trend.

Question 3

For this question, use the monthly Australian short-term overseas visitors data, May 1985–April 2005. (Data set: `visitors`.)

(a) Make a time plot of your data and describe the main features of the series.

(b) Split your data into a training set and a test set comprising the last two years of available data. Forecast the test set using Holt-Winters' multiplicative method.

```
#Splitting data
train <- window(visitors, end=end(visitors)-c(2,0))
test <- window(visitors, start = end(visitors) - c(2,-1))

#Forecast test set and plot forecasts
fcast <- hw(train, h=24, seasonal="multiplicative")
autoplot(fcast) +
  autolayer(visitors)
```


(c) Why is multiplicative seasonality necessary here?

(d) Forecast the two-year test set using each of the following methods:

- an ETS model;
- an ETS model applied to a Box-Cox transformed series;
- a seasonal naive method;
- an STL decomposition applied to the Box-Cox transformed data followed by an ETS model applied to the seasonally adjusted (transformed) data.

```
#Forecasts
f1 <- forecast(ets(train), h = 24)
      #Alternatively, ets(train) %>% forecast(h = 24)
```

```
f2 <- forecast(ets(train, lambda=0), h = 24)

f3 <- snaive(train, h = 24)

f4 <- stlf(train, lambda = 0, etsmodel = "ZZN", h = 24)
                              #Why 'N' for Seasonal component?

#Print output
f1
f2
f3
f4

#Plot forecasts
autoplot(visitors) +
  autolayer(f1, PI=FALSE, series="ETS") +
  autolayer(f2, PI=FALSE, series="ETS with Box-Cox") +
  autolayer(f3, PI=FALSE, series="Seasonal naive") +
  autolayer(f4, PI=FALSE, series="STL+ETS with Box-Cox")
```


(e) Which method gives the best forecasts? Does it pass the residual tests?

(f) Compare the same four methods using time series cross-validation with the `tsCV` function instead of using a training and test set. Do you come to the same conclusions?

- Recall the usage of `tsCV`: `tsCV(y, forecastfunction, h = 1, window = NULL,...)`.

- The second argument requires us to specify the forecast function.

- Since `snaive` and `stlf` are inbuilt forecast functions, we can specify them directly for the `forecastfunction` argument. That is, `tsCV(visitors, forecastfunction = snaive, ...)` and `tsCV(visitors, forecastfunction = stlf,...)`.
- However, we will have to write the forecast function for the ETS models that will then enter as the argument for `forecastfunction`. `ets()` does not produce forecasts for us - it only selects the ETS components and estimates the corresponding parameters of the models.

- So lets write a forecast function that produces forecasts for the ETS model:

```
#Lets call the function f1

f1 <- function(y, h) {
  forecast(ets(y), h = h)
}
```


Subsequently, we can use the `tsCV()` function, save the subsequent forecast residuals and calculate the mean squared error (Recall that the `tsCV()` function returns a vector of forecast errors):

```
e1 <- tsCV(y = visitors, forecastfunction = f1, h = 1)
mean(e1^2, na.rm = T)
```

- Lets now write the forecast function that produces forecasts for the ETS model applied to a Box-Cox transformed series:

```
#Lets call the function f2

f2 <- function(y, h) {
  forecast(ets(y, lambda = 0), h = h)
}
```

Subsequently, we can use the `tsCV()` function, save the subsequent forecast residuals and calculate the mean squared error (Recall that the `tsCV()` function returns a vector of forecast errors):

```
e2 <- tsCV(y = visitors, forecastfunction = f2, h = 1)
mean(e2^2, na.rm = T)
```

We can now proceed with time series cross-validation with the seasonal naive and STLF functions and calculate their respective mean squared error:

```
e3 <- tsCV(visitors, forecastfunction = snaive)
e4 <- tsCV(visitors, forecastfunction = stlf, lambda=0, h = 1)

mean(e3^2, na.rm = T)
mean(e4^2, na.rm = T)
```


Now the STLF method appears better (based on 1-step forecasts), even though it was worst on the test set earlier.

Question 4

The `fets()` function below returns ETS forecasts.

```
fets <- function(y, h) {
  forecast(ets(y), h = h)
}
```

(a) Apply `tsCV()` for a forecast horizon of $h = 4$, for both ETS and seasonal naive methods to the `qcement` data. Do so by using the newly created `fets()` and the existing `snaive()` functions as your forecast function arguments. Recall that the `tsCV()` function returns a vector of forecast errors.

```r
e1 <- tsCV(qcement, fets, h=4)
e2 <- tsCV(qcement, snaive, h=4)
```

(b) Compute the MSE of the resulting 4-step-ahead errors. (Hint: make sure you remove missing values.) Comment on which forecasts are more accurate. Is this what you expected?

```r
colMeans(e1^2, na.rm=TRUE)
colMeans(e2^2, na.rm=TRUE)
```