Dynamic Models for Prediction

Tutorial 2

Simple exponential smoothing

- Plot the simple exponential smoothing to the oil production in Saudi Arabia for three different alpha={0.2,0.6} and an estimated alpha (optimum)?
  - For the first two cases the initial state values should be based on the first few observations but for the last case we want to get the optimum initial state value.

Simple exponential smoothing

1. Lets restrict our time window

```
oildata <- window(oil, start = 1996, end = 2007)
```

and plot

```
plot(oildata, ylab = "Oil (millions of tonnes)", xlab = "Year")
```

2. Apply the SES with a={0.2,0.6} and initial state value based on first observations

```
fit1 <- ses(oildata, alpha = 0.2, initial = "simple", h = 3)
fit2 <- ses(oildata, alpha = 0.6, initial = "simple", h = 3)
```

3. Apply the SES optimizing the alpha and the initial state value by minimizing SSE

```
fit3 <- ses(oildata, h = 3)
```

To see the optimum values considered by ses: fit3$model

•

Simple exponential smoothing

```
plot(fit1, plot.conf=FALSE, ylab="Oil (millions of tonnes)",
    xlab="Year", main="", fcol="white", type="o")
lines(fitted(fit1), col="blue", type="o")
lines(fitted(fit2), col="red", type="o")
lines(fitted(fit3), col="green", type="o")
lines(fit1$mean, col="blue", type="o")
lines(fit2$mean, col="red", type="o")
lines(fit3$mean, col="green", type="o")
legend("topleft",lty=1, col=c(1,"blue","red","green"), c("data", expression(alpha
== 0.2), expression(alpha == 0.6),expression(alpha == 0.89)),pch=1)
```

Forecasting of data with a trend

For the air passenger's data, with a=0.8 and b=0.2

air <- window(ausair, start = 1990, end = 2004)

- The holt's linear trend method

fit1 <- holt(air, alpha = 0.8, beta = 0.2, initial = "simple", h = 5)

- The exponential trend method

fit2 <- holt(air, alpha = 0.8, beta = 0.2, initial = "simple", exponential = TRUE, h = 5)

- The damped trend method

fit3 <- holt(air, alpha = 0.8, beta = 0.2, damped = TRUE, initial = "simple", h = 5)

Forecasting of data with a trend

```
plot(fit2, type = "o", ylab = "Air passengers in Australia (millions)", xlab = "Year",
  fcol = "white", plot.conf = FALSE)
lines(fitted(fit1), col = "blue")
lines(fitted(fit2), col = "red")
lines(fitted(fit3), col = "green")
lines(fit1$mean, col = "blue", type = "o")
lines(fit2$mean, col = "red", type = "o")
lines(fit3$mean, col = "green", type = "o")
legend("topleft", lty = 1, col = c("black", "blue", "red", "green"),
       legend=c("Data", "Holt's linear trend", "Exponential trend", "Additive damped trend"))
```

Comparing the forecasting performance

Taking the sheep example and assuming that data from 1970-2005 is the training set and data from 2001-2005 the test set.

- The parameters and the initial values are estimated by minimizing SSE.

```
livestock2 <- window(livestock, start = 1970, end = 2000)

fit1 <- ses(livestock2)

fit2 <- holt(livestock2)

fit3 <- holt(livestock2, exponential = TRUE)

fit4 <- holt(livestock2, damped = TRUE)

fit5 <- holt(livestock2, exponential = TRUE, damped = TRUE)
```

Comparing the forecasting performance

- For the SES model:

```
fit1$model
accuracy(fit1) # training set
accuracy(fit1,livestock) # test set
fit1$model$par
```

- Can you find the results for the rest of the models?

- Can you plot all the models together with the data and the forecasts for the next 5 years?

Comparing the forecasting performance

```
plot(fit3, type="o", ylab="Livestock, sheep in Asia (millions)",
    flwd=1, plot.conf=FALSE)
lines(window(livestock,start=2001),type="o")
lines(fit1$mean,col=2)
lines(fit2$mean,col=3)
lines(fit4$mean,col=5)
lines(fit5$mean,col=6)
legend("topleft", lty=1, pch=1, col=1:6,
      c("Data","SES","Holt's","Exponential","Additive
Damped","Multiplicative Damped"),cex=0.5)
```

Holt-Winters seasonal method

Recall that:

- Additive method is preferred when the seasonal variations are roughly constant over the series
- Multiplicative method is preferred when seasonal variations are changing proportional to the level of the series.

Holt-Winters seasonal method

Forecasting the international visitor nights in Australia, for both additive and multiplicative seasonality

```
aust <- window(austourists, start=2005)

plot(aust)

fit1 <- hw(aust,seasonal="additive")

fit2 <- hw(aust,seasonal="multiplicative")

plot(fit2,ylab="International visitor night in Australia (millions)",
 plot.conf=FALSE,type="o",fcol="white",xlab="Year")

lines(fitted(fit1),col="red",lty=2)

lines(fitted(fit2),col="green",lty=2)

lines(fit1$mean,type="o",col="red")

lines(fit2$mean,type="o",col="green")

legend("topleft",lty=1,pch=1,col=c(1,2,3),c("data","Holt Winters' Additive","Holt Winters' Multiplicative"),cex=0.5)
```

Holt-Winters seasonal method

Estimated components for Holt-Winters method with additive and multiplicative seasonal components

```
states <- cbind(fit1$model$states[,1:3],fit2$model$states[,1:3])

colnames(states) <-
c("level","slope","seasonal","level","slope","seasonal")

plot(states,main="Additive  Seasonality
Multiplicative  Seasonality",

    xlab="Year")
```

ETS Models

- Recall the parameters of the state space model ETS(Error,Trend,Seasonal).
- In R the function is ets().
- If you only define the time series then an appropriate model is specified
- Type **help(ets)** and take a look at the parameters of the ets() function.

Forecasting with ETS models

- By passing as argument the object returned by the ets() function to the forecast() function we are able not only to forecast the values of the time series but also the prediction intervals
- Take a look at **forecast.ets()** function parameters

Forecasting with ETS models

- Back to the oil production example

```
oildata <- window(oil, start = 1996, end = 2007)

fit <- ets(oildata, model = "ANN")

plot(forecast(fit, h=3), ylab="Oil (millions of tones)")

summary(fit)

ls(fit)  #list names of the objects in the specified environment

fit$par
```

Forecasting with ETS models

- Back to the international tourist visitor nights in Australia

```
vndata <- window(austourists, start = 2005)
fit <- ets(vndata)
summary(fit)
```

Which is the selected model?

- Plot the fitted model and the forecast for h=8

```
plot(fit)
plot(forecast(fit, h = 8), ylab = "International visitor night in Australia (millions)")
```

For which prediction intervals it was plots?

Example 1

Use the quarterly UL passenger vehicle production data from 1977:1-2005:1 (data set ukcars)

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

b) Decompose the series using STL and obtain the seasonally adjusted data.

c) Forecast the next two years of the series using Holt's linear trend method applied to the seasonally adjusted data.

Reseasonalize the forecasts.

Record the parameters of the method and report the RMSE of the one-step forecasts from your method.

Example 1

d) Forecast the next two years of the series using Holt's linear method applied to the seasonally adjusted data.

Then reseasonalize the forecasts.

Record the parameters of the method and report the RMSE of

the one-step forecasts from your method.

e) Now use ets() to choose a seasonal model for the data.

f) Compare the RMSE of the fitted model with the RMSE of the model you obtained using an STL decomposition with Holt's method. Which gives the better in-sample fits?

g) Compare the forecasts from the two approaches? Which seems most reasonable?

Stationarity

- The properties of a stationary time series do not depend on the time at which the series is observed.
- Stationary time series:
  - have no predictable patterns in the long-term
  - they will show to be roughly horizontal with constant variance in time plots.
  - some cyclic behavior is possible to exist.

Stationarity

Execute the following code. Can you identify which series are stationary?

```
par(mfrow = c(3, 3))
plot(dj, main = "(a)", xlab = "Day")
plot(diff(dj), main = "(b)", xlab = "Day")
plot(strikes, main = "(c)", xlab = "Year")
plot(hsales, main = "(d)", xlab = "Year")
plot(eggs, main = "(e)", xlab = "Year")
plot(pigs, main = "(f)", xlab = "Year")
plot(lynx, main = "(g)", xlab = "Year")
plot(beer, main = "(h)", xlab = "Year")
plot(elec, main = "(i)", xlab = "Year")
```

Differencing

- By computing the differences between consecutive observations you can make a time series stationary. Why? Because:
  - You are stabilizing the mean of a time series by removing the changes in the level of a time series

- Why is it useful to make a time series stationary?
  - To eliminate trend and seasonality

How can you identify a non-stationary time series?

- Apply the ACF, if:
  - The ACF drops quickly to zero → stationary series
  - The ACF decreases slowly → non-stationary series
  - The value of r_1 is large and positive → strong indication of non-stationary series

How can you identify a non-stationary time series?

- Plot the ACF of the Dow-Jones index and the daily changes in the Dow-Jones index

  ```
  par(mfrow = c(1, 2))
  Acf(dj)
  Acf(diff(dj))
  ```

  What can you say about the stationarity of the data?

- Apply the Ljung-Box test and examine the null hypothesis of independence for the 10th lag for diff(dj)

  ```
  Box.test(diff(dj), type = "L", lag = 10)
  ```

  Are the data uncorrelated?

Seasonal differencing

- If first differencing would give you stationary data, then a random walk model would be suitable and naïve forecasts would be done.
- Now if data are still non-stationary after differencing maybe a seasonal differencing should have been done.
- Plot the seasonal difference of the logarithm of the monthly scripts for a10

```
par(mfrow=c(3,1))

plot(a10,xlab="Year")

plot(log(a10),xlab="Year")

plot(diff(log(a10),12), xlab="Year",ylab="Annual change in monthly log A10 sales")
```

Seasonal differencing
+
First differencing

Plot the US net electricity generation, the log of it, the seasonal difference and then apply a first difference on the data

```
par(mfrow=c(2,2))

plot(usmelec)

plot(log(usmelec))

plot(diff(log(usmelec),12))

plot(diff(diff(log(usmelec),12),1))
```

Example 2

- For the following series, find an appropriate Box-Cox transformation and order of differencing in order to obtain stationary data.
    - (a) usnetelec
    - (b) usgdp