Time Series Analysis & Forecasting

Class 3

Arnab Bose, Ph.D.

MSc Analytics

University of Chicago

Data Transformations

- Log transform
- Box-Cox transformation w_t of TS y_t

$$w_t = \begin{cases} \log(y_t) \;\; if\ \lambda = 0 \\ \dfrac{y_t^{\lambda} - 1}{\lambda} \; otherwise \end{cases}$$

- Use a transformation to
  - Decouple mean and variance to remove variance dependence on mean
  - Model is simple (additive)
  - Residuals are more or less normally distributed with zero mean and constant variance

R code

- *library("forecast", lib.loc="~/R/win-library/3.3")*
- *lambda <- BoxCox.lambda(lynx)*

Wold Decomposition

A (weak) stationary TS can be written as the sum of 2 TS, one deterministic and one stochastic

$$Y_t = \sum_{i=0}^{\infty} \psi_i w_{t-i} + \eta_t$$

Where

η_t is deterministic process

$\psi_0 = 1$ and $\sum_{i=0}^{\infty} {\psi_i}^2 < \infty$ (process is stable)

w_t is white noise

$Cov(w_t, \eta_t) = 0$ for all t, s

Simple Operators

- Backward shift operator $B\, z_t = z_{t-1}$
- Forward shift operator $F\, z_t = z_{t+1}$
- Backward difference operator $\nabla\, z_t = z_t - z_{t-1}$

Linear Process

- TS $\{y_t\}$ is a weighted linear representation of independent shocks $\{e_t\}$ that represent white noise

- Assume $\{e_t\}$ to be normally distributed with mean 0 and variance σ_e^2

- Mathematically the TS can be represented as

$$y_t = \mu_t + e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \ldots$$

where

μ_t - determines the "level" of the process

ψ_t - constant coefficients that need to be summable for stationarity

Autoregressive (AR)

- Current value of a process is expressed as a finite, linear aggregate of the pervious values of the process and white noise $\{e_t\}$

- Mathematically AR(p): $Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + e_t$

- Define operator $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots . - \phi_p B^p$

- Mathematically the TS can be represented as

$$\Phi(B) Y_t = e_t$$

AR Stationarity

- Mathematically the TS can be represented as

$$\Phi(B)Y_t = e_t$$

$$Y_t = \Phi^{-1}(B)e_t = \Psi(B)e_t$$

- For stationarity, $\Psi(B)$ needs to be a convergent series

- In other words, all roots of $\Phi(B) = 0$ must be greater than 1 in absolute value => all roots must lie outside the unit circle

Moving Average (MA)

- Current value of a process depends on a finite number of white noise $\{e_t\}$

- Mathematically MA(q): $Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} + \ldots + \theta_q e_{t-q}$

- Define operator $\Theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots . - \theta_q B^q$

- Mathematically the TS can be represented as

$$Y_t = \Theta(B) e_t$$

MA Invertibility

- MA process has the characteristic polynomial

$$\Theta(x) = 1 - \theta_1 x - \theta_2 x^2 - \ldots - \theta_q x^q$$

- And the characteristic equation

$$1 - \theta_1 x - \theta_2 x^2 - \ldots - \theta_q x^q = 0$$

- MA(q) is invertible if and only if the roots of the characteristic equation > 1, i.e. all roots must lie outside the unit circle

Mixed Autoregressive Moving Average Model

- Assume TS is partly AR(p) and partly MA(q)

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} + \ldots + \theta_q e_{t-q}$$

- ARMA(p,q) mathematically represented as

$$\Phi(B) Y_t = \Theta(B) e_t$$

- ARMA(p,q) require both stationarity and invertibility

Models for Non-stationary TS

- Consider a Random Walk model $Y_t = Y_{t-1} + w_t$
- Take first difference $Y_t - Y_{t-1} = w_t$ => gives white noise that is stationary
- Represented as $\nabla Y_t = w_t$
- d^{th} difference to get process stationarity and then apply ARMA(p,q) process => ARIMA
- ARIMA(p,d,q) mathematically represented as

$$\Phi(B)(1-B)^d Y_t = \Theta(B)e_t$$

Model Specification

- For MA(q) TS, autocorrelation $\rho_k = 0, k > q$

- For AR(p) TS, partial autocorrelation $\phi_{kk} = 0, k > q$

- Akaike Information Criterion $AIC = 2k - 2\ln(L)$
  where
  - k – # of parameters in the model
  - L – maximum value of the likelihood estimator

- Corrected Akaike Information Criterion $AICc = AIC + \frac{2k(k+1)}{n-k-1}$, n is the sample size

- Bayesian Information Criterion $BIC = -2\ln(L) + k\ln(n)$

R code

- *ar1 <- arima.sim(list(order=c(1,0,0), ar=0.75), n=100)*
- *acf(ar1)*
- *pacf(ar1)*

- *ma1 <- arima.sim(list(order=c(0,0,1), ma=0.9), n=100)*
- *acf(ma1)*
- *pacf(ma1)*

- *arma11 <- arima.sim(list(order=c(1,0,1), ar=0.75, ma=0.9), n=100)*
- *acf(arma11)*
- *pacf(arma11)*
- *eacf(arma11)*

- *arma21 <- arima.sim(list(order=c(2,0,1), ar=c(0.9, -0.75), ma=0.9), n=100)*
- *acf(arma21)*
- *pacf(arma21)*
- *eacf(arma21)*

R code

- *lynx.fit.arima.boxcox <- auto.arima(lynx, lambda=lambda) # Using Box-Cox transformation*
- *plot(forecast(lynx.fit.arima.boxcox, h=20))*

- *# note if you do the Box-Cox separately, then forecast does not invert it*
- *lambda <- BoxCox.lambda(lynx)*
- *lynx.fit.arima <- auto.arima(BoxCox(lynx, lambda))*
- *plot(forecast(lynx.fit.arima, h=20))*

R code (continued)

- *ar_m <- arima.sim(list(order=c(1,0,0), ar=0.75), n=100) + 10 # adding mean to AR process*
- *plot(ar_m)*
- *mean(ar_m)*

- *ar_im <- arima.sim(list(order=c(1,0,0), ar=0.75), n=100, mean = 10) # adding mean to innovation (error)*
- *plot(ar_im)*
- *mean(ar_im)*

- *ma_im <- arima.sim(list(order=c(0,0,1), ma=0.9), n=100, mean = 10) # adding mean to innovation (error)*
- *plot(ma_im)*
- *mean(ma_im)*

Textbook Chapters

- Materials covered:
  - FPP: Chapter 8, MKJ: Chapter 5, TSA: Chapters 4 – 6