Time Series Analysis & Forecasting

Class 8

Arnab Bose, Ph.D.

MSc Analytics

University of Chicago

Intervention Analysis

Framework to study effect of intervention on a TS

$$Y_t = m_t + N_t$$

where m_t : change in the mean function

N_t : ARIMA process without intervention

2 Types of Intervention at time *T*:

Step function $$S_t^{(T)} = \begin{cases} 1, & t \geq T \\ 0, & otherwise \end{cases}$$

Pulse function $$P_t^{(T)} = \begin{cases} 1, & t = T \\ 0, & otherwise \end{cases}$$

Intervention causes change in mean defined by Type equation here.

Intervention Analysis

Intervention causes change in mean defined by

$$m_t = \omega S_t^{(T)}$$

Where ω is the unknown permanent change due to the intervention

Intervention with delay of *d* time units is defined by

$$m_t = \omega S_{t-d}^{(T)}$$

Intervention may affect the mean function gradually and can be modeled as AR(1)

$$m_t = \delta m_{t-1} + \omega S_{t-1}^{(T)}$$

Intervention Analysis

Intervention may affect the mean function gradually and can be modeled as AR(1)

$$m_t = \delta m_{t-1} + \omega S_{t-1}^{(T)}$$

$$m_t = \begin{cases} \omega \dfrac{1-\delta^{t-T}}{1-\delta}, & t > T \\ 0, & otherwise \end{cases}$$

Usually $0 < \delta < 1$ so that the ultimate change in mean for large t is

$$m_t = \frac{\omega}{1-\delta}$$

Intervention Analysis

Likewise short lived intervention is specified as

$$m_t = \delta m_{t-1} + \omega P_{t-1}^{(T)}$$

$$\Rightarrow m_t = \delta B m_t + \omega B P_t^{(T)}$$

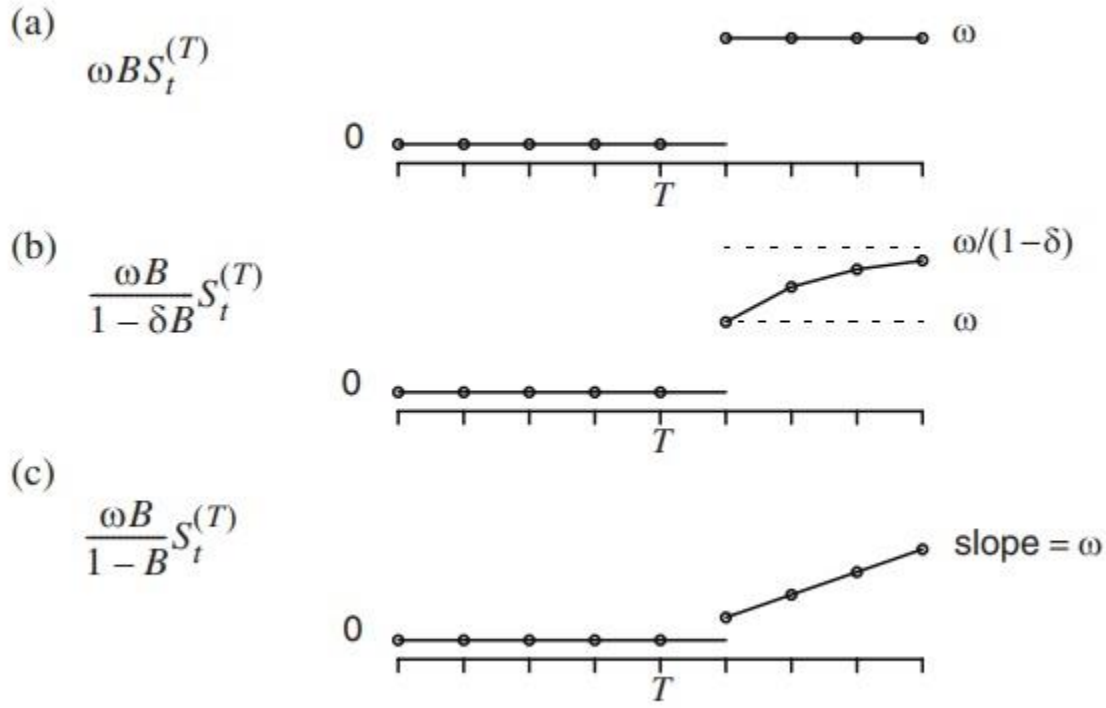
$$\Rightarrow m_t = \frac{\omega B}{1 - \delta B} P_t^{(T)}$$

Note that

$$S_t^{(T)} = \frac{1}{1 - B} P_t^{(T)}$$

Step Interventions

Exhibit 11.3 Some Common Models for Step Response Interventions (All are shown with a delay of 1 time unit)

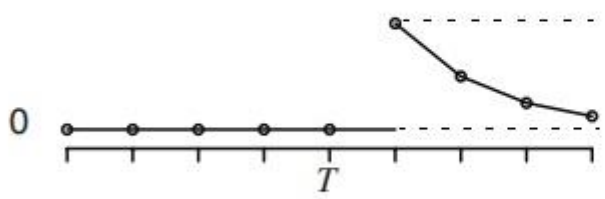


Source: TSA, page 253

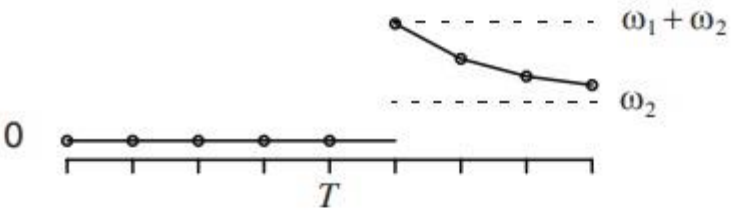
Pulse Interventions

Exhibit 11.4 Some Common Models for Pulse Response Interventions (All are shown with a delay of 1 time unit)

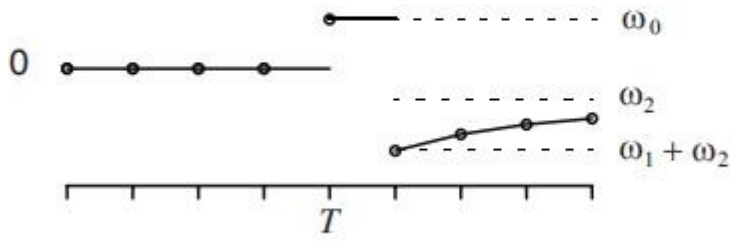
(a) $\frac{\omega B}{1-\delta B} P_t^{(T)}$



(b) $\left[\frac{\omega_1 B}{1-\delta B} + \frac{\omega_2 B}{1-B}\right] P_t^{(T)}$



(c) $\left[\omega_0 + \frac{\omega_1 B}{1-\delta B} + \frac{\omega_2 B}{1-B}\right] P_t^{(T)}$



Source: TSA, page 254

R code – Intervention Analysis

- *library("TSA", lib.loc="~/R/win-library/3.2")*
- *data(airmiles)*
- *I911 <- 1*(seq(airmiles)==69)*
- *S911 <- 1*(seq(airmiles)>=69)*

- *air.mPulse <- arimax(log(airmiles),order=c(0,1,1),seasonal=list(order=c(0,1,1), period=12),xtransf=data.frame(I911, I911), transfer=list(c(0,0),c(1,0)), method='ML')*
- *air.mPulse*

Coefficients:

	ma1	sma1	I911-MA0	I911.1-AR1	I911.1-MA0
	-0.5379	-0.7644	-0.1290	0.8901	-0.2419
s.e.	0.0854	0.1532	0.0606	0.1239	0.0513

sigma^2 estimated as 0.0009739: log likelihood = 199.42, aic = -388.84

- *plot(ts(filter(I911, filter=0.8901, method='recursive', side=1)*(-0.2419), frequency = 12, start=1996), type='h',ylab='9/11 Pulse Effects')*

R code – Intervention Analysis

- *air.mStep <- arimax(log(airmiles),order=c(0,1,1),seasonal=list(order=c(0,1,1), period=12),xtransf=data.frame(S911, S911), transfer=list(c(0,0),c(1,0)), method='ML')*
- *air.mStep*

Coefficients:

	ma1	*sma1*	*S911-MA0*	*S911.1-AR1*	*S911.1-MA0*
	-0.5174	*-0.7710*	*2.2415*	*-0.0618*	*-2.6169*
s.e.	*0.0850*	*0.1444*	*1.5261*	*0.0356*	*1.5303*

sigma^2 estimated as 0.001034: log likelihood = 196.3, aic = -382.6

- *plot(ts(S911*(2.2415)+filter(S911, filter=-0.0618, method='recursive', side=1)*(-2.6169), frequency = 12, start=1996), type='h',ylab='9/11 Step Effects')*

Forecasting using arimax( )

1. arimax( ) does not implement a predict( ) function => cannot use forecast( )

2. Workaround from https://stats.stackexchange.com/questions/169564/arimax-prediction-using-forecast-package :
    1. Use the forecast package arima function to determine the pre-intervention noise series and add any outlier adjustment.
    2. Fit the same model in arimax but add the transfer function
    3. Take the fitted values for the transfer function (coefficients from arimax) and add them as xreg in arima.
    4. Forecast with arima

3. Code:
    1. *air.m1<-arimax(log(airmiles), order=c(0,1,1), seasonal = c(0,1,1), xtransf=data.frame(I911=1*(seq(airmiles)==69)), transfer=list(c(1,0)))*
    2. *tf<-filter(1*(seq(1:(length(airmiles)+5))==69), filter=0.6948, method='recursive',side=1) * (-0.3459)*
    3. *forecast.arima<-Arima(log(airmiles), order=c(0,1,1), seasonal = c(0,1,1), xreg=tf[1:(length(tf)-5)])*
    4. *forecast.arima*
    5. *predict(forecast.arima, n.ahead = 5, newxreg=tf[114:length(tf)])*

Outlier Analysis – General

tsoutliers and tsclean: https://robjhyndman.com/hyndsight/forecast5/

- *y<-c(0.59, 0.61, 0.59, 1.55, 1.33, 3.50, 1.00, 1.22, 2.50, 3.00, 3.79, 3.98, 4.33, 4.45, 4.59, 4.72, 4.82, 4.90, 4.96, 7.92, 5.01, 5.01, 4.94, 5.05, 5.04, 5.03, 5.06, 5.10, 5.04, 5.06, 7.77, 5.07, 5.08, 5.08, 5.12, 5.12, 5.08, 5.17, 5.18)*
- *ts.plot(y)*
- *tsoutliers(y)*

- *y_clean <- tsclean(y)*
- *lines(y_clean)*

Outlier Analysis – Detailed

Outliers happen due to measurement and/or copying errors or abrupt changes to the underlying process

2 Types of Outliers:

Additive Outliers

$$Y_t' = Y_t + \omega_A P_t^{(T)}$$

where Y_t is the unperturbed TS

Innovative Outliers occurs at time *t* if the error is perturbed

$$e_t' = e_t + \omega_I P_t^{(T)}$$

R code – Additive Outlier

- *library("TSA", lib.loc="~/R/win-library/3.2")*
- *set.seed(12345)*
- *y <- arima.sim(model=list(ar=0.8, ma=0.5), n.start=158, n=100)*
- *y[10]*
- *ts.plot(y)*
- *y[10] <- 10*
- *ts.plot(y)*
- *acf(y)*
- *pacf(y)*
- *eacf(y)*
- *m1 <- Arima(y, order=c(1,0,0))*
- *m1*
- *detectAO(m1)*
- *detectAO(m1)*
- *m2 <- Arima(y, order=c(1,0,0), xreg=data.frame(AO=seq(y)==10))*
- *m2*

R code – Innovation Outlier

- *#co2 example from TSA*
- *data(co2)*
- *m1.co2 <- Arima(co2, order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12))*
- *m1.co2*
- *detectAO(m1.co2)*
- *detectIO(m1.co2)*
- *m1.co2.io <- arimax(co2, order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12),io=c(57))*
- *acf(m1.co2$residuals)*
- *acf(m1.co2.io$residuals)*

ARCH Model

Have a TS $\{\varepsilon_t\}$ with heteroscedasticity.

ARCH(q) model represented as

$$\varepsilon_t = \sigma_t z_t$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-2}^2 + .. + \alpha_q \varepsilon_{t-q}^2$$

ARCH Model

ARMA (p,q) process

$$\phi(B)Y_t = \theta(B)e_t$$

where e_t is iid random noise

$\{e_t\}$ is a TS with heteroscedasticity.

Detect heteroscedasticity:

McLeod Li test for Box-Jenkins models:

Ljung Box statistic for the squared TS

$$H_0: no\ ARCH\ effect\ in\ data$$

Breusch-Pagan test for linear regression model:

Model with n parameters:

$$H_0: homoscedastic\ \chi^2\ n-1\ distribution$$

ARCH Model

The conditional mean of r_t given past information

$$E[r_t|r_{t-1}, r_{t-2}, \dots] = E[\sigma_t z_t|r_{t-1}, r_{t-2}, \dots]$$
$$= \sigma_t E[z_t|r_{t-1}, r_{t-2}, \dots]$$
$$= \sigma_t * 0 = 0$$

ARCH(1) process is serially uncorrelated

$$E[r_t r_{t-1}] = E[E[r_t r_{t-1}|r_{t-1}, r_{t-2}, \dots]]$$
$$= E[r_{t-1} E[r_t|r_{t-1}, r_{t-2}, \dots]]$$
$$= E[r_{t-1} * 0] = 0$$

Likewise $Cov[r_t r_{t-1}] = 0$, in other words r_t cannot be predicted using historical information => evidence for Efficient Market Hypothesis

ARCH Model

However, r_t^2 can be predicted given past information

$$\begin{aligned} Var[r_t|r_{t-1}, r_{t-2}, \dots] &= E[r_t^2|r_{t-1}, r_{t-2}, \dots] \\ &= E[\sigma_t^2 z_t^2|r_{t-1}, r_{t-2}, \dots] \\ &= \sigma_t^2 E[z_t^2|r_{t-1}, r_{t-2}, \dots] \\ &= \sigma_t^2 * 1 = \sigma_t^2 \end{aligned}$$

GARCH Model

Generalized ARCH

GARCH(p, q) model is represented as

$$\varepsilon_t = \sigma_t z_t$$

$$\sigma_t^2 = \alpha_0 + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2 + .. + \beta_p \sigma_{t-p}^2 + \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-2}^2 + .. + \alpha_q \varepsilon_{t-q}^2$$

GARCH to ARCH Model

GARCH(p, q) model is represented as

$$\varepsilon_t = \sigma_t z_t$$

$$\sigma_t^2 = \alpha_0 + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2 + .. + \beta_p \sigma_{t-p}^2 + \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-2}^2 + .. + \alpha_q \varepsilon_{t-q}^2$$

$$\sigma_t^2 = \alpha_0 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 + \sum_{j=1}^{q} \alpha_j \varepsilon_{t-j}^2$$

GARCH(1,1) model

$$\sigma_t^2 = \alpha_0 + \beta_1 \sigma_{t-1}^2 + \alpha_1 \varepsilon_{t-1}^2$$

Equivalent to ARCH(k) model (k is very large)

$$\sigma_t^2 = \frac{\alpha_0}{1-\beta} + \alpha_1 \sum_{j=1}^{k} \beta^{j-1} \varepsilon_{t-j}^2$$

R code – Garch

- *library("rugarch", lib.loc="~/R/win-library/3.6")*
- *data("sp500ret")*
- *acf(sp500ret, lag = 50)*
- *acf(sp500ret^2, lag = 50)*
- *McLeod.Li.test(y=sp500ret$SP500RET)*
- *modelSpec<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),*
- *mean.model = list(armaOrder = c(1,1), include.mean = T), distribution.model = "std")*
- *modelSpec@model$pars*
- *model<-ugarchfit(spec=modelSpec,data=sp500ret)*
- *coef(model)*
- *plot(model@fit$sigma^2)*
- *plot(model)*
- *# News Impact Curve – a handy tool employed by Pagan and Schwert (1990) and Engle and Ng (1993) to compare the various asymmetric garch models. It plots the relation between the conditional volatility σ_t and the shocks ε_t and is used to compare the various asymmetric volatility models and conclude that the models differ in the way they accommodate asymmetry. To explain this phenomenon, the return shocks or error term of the mean equation ε_t is categorized as collectively measuring news. So a positive ε_t an unexpected increase in price – suggests the arrival of some good news, while a negative ε_t – an unexpected decrease in price – denotes the arrival of some bad news.*

Different GARCH Models

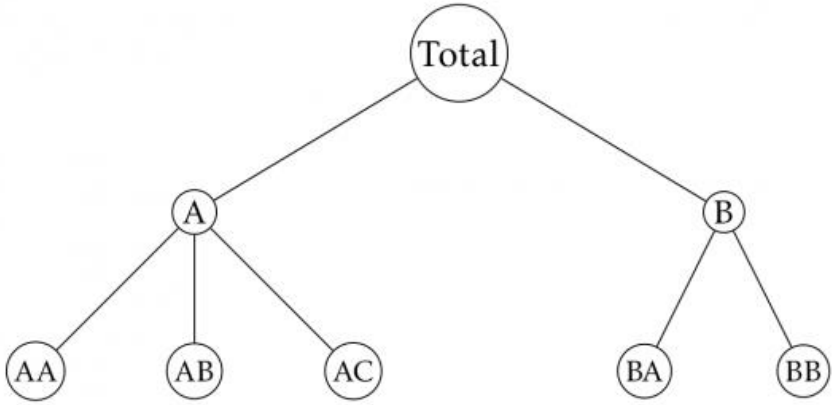
IGARCH Integrated GARCH – to model persistent changes in volatility

APARCH Asymmetric Power ARCH – no symmetric square values effect

NGARCH Non-linear GARCH

EGARCH Exponential GARCH

Hierarchical Time Series

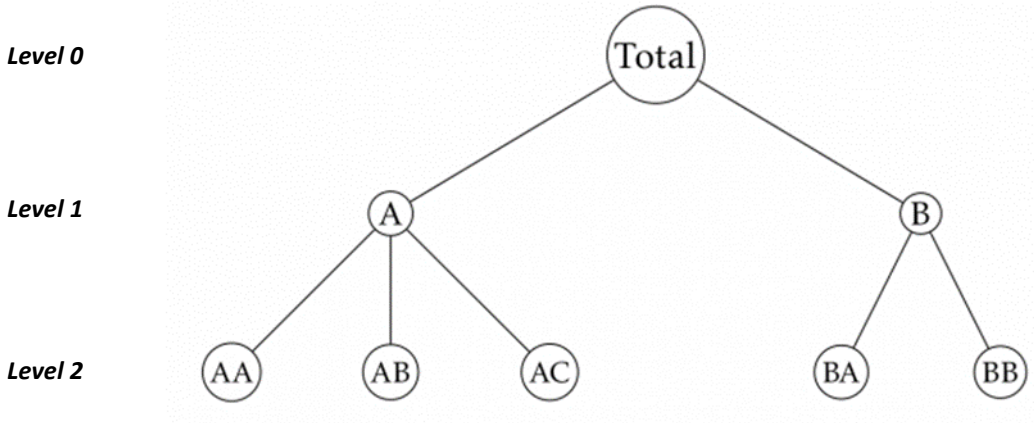


A two level hierarchical tree diagram*

- Bottom-up approach – generate forecasts for the bottom level TS and keep aggregating up the hierarchy. Pro: no loss of info. Con: noisy data that is challenge to model and forecast
- Top-down approach – generate forecasts for the "Total" series at the top and then disaggregate downward using a set of proportions to distribute to each series at the lower levels. Pro: simple to model. Con: unable to capture individual TS
- Middle-out approach – chose a "middle level" and forecast for all TS at this level. For TS above, aggregate like bottom-up. For series below, disaggregate like top-down.

* https://www.otexts.org/fpp/9/4 , Accessed May 2018

R code – Hierarchical Time Series




```
library("hts", lib.loc="~/R/win-library/3.3")
nodes <- list(2, c(3, 2))
abc <- ts(5 + matrix(sort(rnorm(500)), ncol = 5, nrow = 100))
x <- hts(abc, nodes)
summary(x)
smatrix(x)
plot(x)
fcst <- forecast(x, method="mo", fmethod = “arima”, level = 1, h=10)
plot(fcst)
```

R hts package – arima model params ?

- hts forecasting does not output the arima (P, D, Q)
- Because a separate ARIMA model is estimated for every series in the hierarchy. For large hierarchies, that can involve thousands or even millions of models. There is no point in storing all the resulting information when you only want forecasts.
- If you want the individual models, then fit them explicitly to all series. You can get the matrix of every series (included aggregated series) using *aggts()*:

```
y <- aggts(fcst)

models <- list()

for(i in 1:ncol(y))

    models[[i]] <- auto.arima(y[,i])
```

Mixed-effects Regression Model (MRM)

- Consider a simple linear regression to time series data for different individuals $i = 1,2, \dots, N$ and at different time points $j = 1,2, \dots, n_i$

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + e_{ij}$$

- e_{ij} assumed to be normal and iid – not a reasonable assumption since same subject errors will have some correlations
- β_0 and β_1 same implies change across individuals do not vary- not reasonable so need to add individual-specific effects into the model
- MRMs – view as augmented linear regression models
- MRM models as 2 levels
  - Level 1 – within-subject variations

$$y_{ij} = b_{0i} + b_{1i} x_{ij} + e_{ij}$$

  - Level 2 – between-subject variations

$$b_{0i} = \beta_0 + v_{0i}$$
$$b_{1i} = \beta_1 + v_{1i}$$

- Two random individual specific effects – v_{0i} and v_{1i}
- $v_{0i} = 0$ – the random intercept model where each individual has their own intercept
- The random effects variance-covariance matrix

$$\Sigma_v = \begin{bmatrix} \sigma_{v_0}^2 & \sigma_{v_0 v_1} \\ \sigma_{v_0 v_1} & \sigma_{v_1}^2 \end{bmatrix}$$

- $\sigma_{v_0 v_1}$ - covariance between v_{0i} and v_{1i}

R code – MRM

- *library("lme4", lib.loc="~/R/win-library/3.2")*
- *data("sleepstudy")*
- *xtabs(Reaction ~ Subject + Days, sleepstudy)*

- *mrm_corr <- lmer(Reaction ~ 1 + Days + (1 + Days|Subject), sleepstudy, REML = 0)*
- *ranef(mrm_corr)[["Subject"]]*

- *mrm_no_corr <- lmer(Reaction ~ 1 + Days + (1|Subject) + (0+Days|Subject), sleepstudy, REML = 0)*

Textbook Chapters

- Materials covered available in book:
  - MJK: Chapter 7, TSA: Chapters 11 – 12

- https://otexts.com/fpp2/hts.html