Date: Thursday, November 15, 2018

Addressing Seasonality

$\nabla Y_t = (1-B)Y_t = Y_t - Y_{t-1}$

Ordinary differencing doesn't work to remove seasonal effect. For this we need seasonal differencing.

(Explain watch sweats again :P)

Notation: $(1-B^k) \equiv \nabla_k$ ← "lag k differencing"

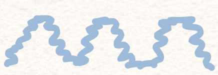
$$\boxed{\nabla_k Y_t = (1-B^k) = Y_t - Y_{t-k}}$$

* This is different from $\nabla^k = (1-B)^k$ which represents k iterations of ordinary differencing.

IDEA: Seasonal effects S_t manifest themselves as $S_t = S_{t+m}$ if the seasonal effect has period m.

For instance, if we form a lag-m difference finitely many times, we can typically remove/mitigate the seasonal effect.

Example: $Y_t = S_t + \varepsilon_t$



$$\begin{aligned}\nabla_m Y_t &= (1-B^m)Y_t \\ &= (1-B^m)\, S_t + \varepsilon_t \\ &= (\cancel{S_t - S_{t-m}}) + (\varepsilon_t - \varepsilon_{t-m}) \\ &= \nabla_m \varepsilon_t.\end{aligned}$$

(Since S_t is a seasonal effect with period m).

So, we hope that after finitely many seasonal differences (and perhaps finitely many ordinary differences) the resultant time series is stationary and hence can be modeled by an ARMA model.

→ season
→ trend

Mathematically, the order of differencing does not matter.

$$\nabla^d \nabla_k^D Y_t = (1-B)^d (1-B^k)^D Y_t$$
$$\nabla_k^D \nabla^d Y_t = (1-B^k)^D (1-B)^d Y_t$$

How do we choose m? The period m will be the number of lags needed for one cycle of the seasonal effect on an ACF plot.

SARIMA "Seasonal ARIMA"

$$\{Y_t\} \sim \text{SARIMA}(p,d,q) \times (P,D,Q)_m$$

if $X_t = (1-B^m)^D (1-B)^d Y_t$ can be modeled by a stationary ARMA model.

$$\phi^*(B) X_t = \theta^*(B) \varepsilon_t$$

$\phi(B)\,\Phi(B^m)$ $\quad\quad$ $\theta(B)\,\Theta(B^m)$

$\{\varepsilon_t\} \sim WN(0,\sigma^2)$

$$\Rightarrow \phi(B)\,\Phi(B^m)(1-B^m)^D(1-B)^d Y_t = \theta(B)\,\Theta(B^m)\,\varepsilon_t$$

where,

$\phi(z) = 1 - \phi_1 z - \phi_2 z^2 - \dots - \phi_p z^p$ ← p^{th} degree polynomial

$\Phi(z^m) = 1 - \Phi_1 z - \Phi_2 z^2 - \dots - \Phi_P z^P$ ← P^{th} degree polynomial

$\theta(z) = 1 + \theta_1 z + \theta_2 z^2 + \dots + \theta_q z^q$ ← q^{th} degree polynomial

$\Theta(z^m) = 1 + \Theta_1 z + \Theta_2 z^2 + \dots + \Theta_Q z^Q$ ← Q^{th} degree polynomial

IDEA: The data between seasons forms a time series and the data within a season forms a time series. These two time series have different ARMA representations.

Example: Suppose $\{Y_t\}$ is recorded quartely and so $m=4$

$$
\begin{array}{cccc}
Y_1 & Y_2 & Y_3 & Y_4 \\
Y_5 & Y_6 & Y_7 & Y_8 \\
Y_9 & Y_{10} & Y_{11} & Y_{12} \\
Y_{13} & Y_{14} & Y_{15} & Y_{16} \\
\vdots & \vdots & \vdots & \vdots
\end{array}
$$

Rows represent within season time series which may be modeled by ARMA (p,q)

Columns represent between season time series which may be modeled by ARMA (P,Q).

* p,q = ARMA order of the within season series.
* P,Q = ARMA order of the between season series.

Order selection:

Step 1: Choose d, m, D such that $X_t = (1-B^m)^D(1-B)^d Y_t$ is stationary.

Step 2: Examine ACF/PACF plots of $\{X_t\}$ to choose p,q,P,Q.

→ p,q are chosen such that $\rho(1), \rho(2), \ldots, \rho(m-1)$ and $\alpha(1), \alpha(2), \ldots, \alpha(m-1)$ are compatible with ARMA (p,q).

→ P,Q are chosen such that $\rho(km)$ and $\alpha(km)$ for $k \in \mathbb{Z}^+$ are compatible with ARMA (P,Q).

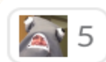
* This procedure can provide a good starting point but optimal orders should be selected via likelihood ratio tests and comparisons of goodness-of-fit metrics.

Box-Jenkins Methodology

1. Check for non-constant variance and apply a transformation if necessary.
2. Check for seasonality and trend and difference as necessary to make stationary.
3. Identify p, q, P, Q from ACF and PACF plots of the (potentially) differenced data. Hence, choose your model.
4. Fit the proposed model and iterate to an optimal one.
5. Check residuals to verify model assumptions. Make adjustments as necessary.
6. Forecast into the future.

[break]

[R]



$$y^* = \begin{cases} \log(y) & \text{if } \lambda = 0 \\ \dfrac{y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \end{cases}$$

Box-Cox transformation!

BoxCox.lambda(data)