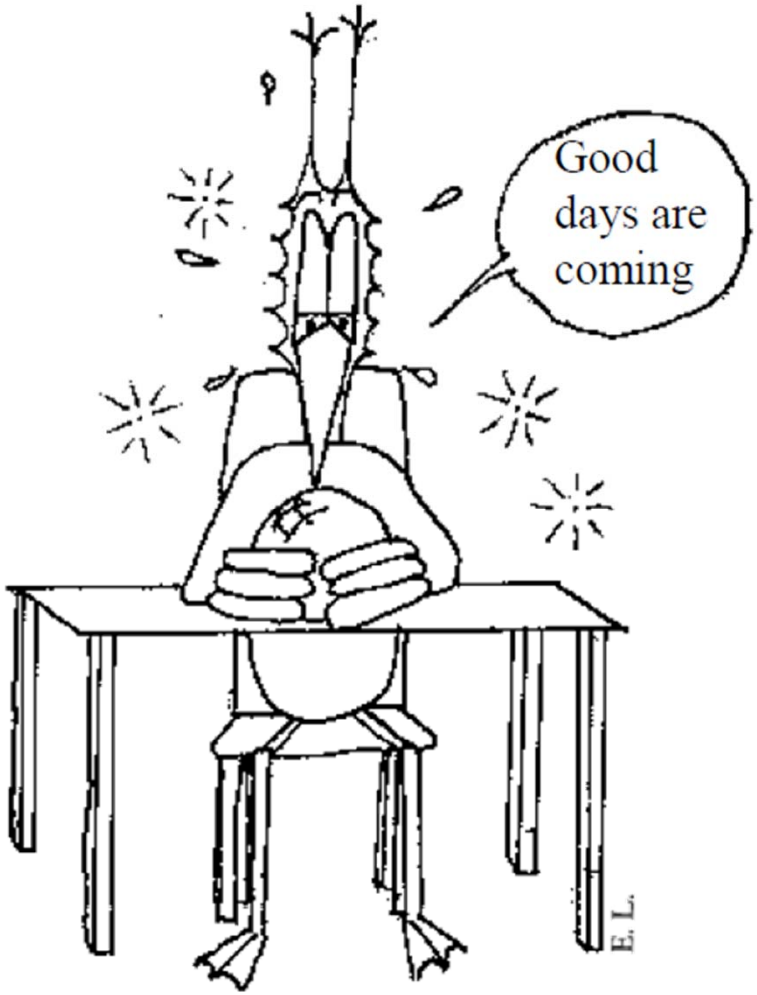
Forecasting
Part 2

JY Le Boudec



Contents

6. Differencing the Data

We have seen that changing the scale of the data may be important for obtaining a good model.

Another kind of pre-processing is the application of *filters*. The idea is that the filter may remove deterministic patterns and it may be simpler to forecast the filtered data

A filter (in full, discrete-time causal filter) is a mapping from the set of finite-length time series to the same set.

By convention, we consider that a filter keeps the length of the time-series unchanged.

Further, a filter has to be linear, time-invariant and causal. The latter means that output of the filter up to time t depends only on the input up to time t.

Differencing filter Δ_1

Differencing filter Δ_1= discrete-time derivative

$$Y = (Y_1, \dots, Y_t) \mapsto \Delta_1 Y = (Y_1, Y_2 - Y_1, \dots, Y_t - Y_{t-1})$$

$$\Delta_1 Y = X \Leftrightarrow X \text{ has same length as } Y \text{ and } X_t = Y_t - Y_{t-1}$$

with the (matlab) convention that $Y_t = 0$ whenever $t \leq 0$

Δ_1 is a filter, thus is linear, $\Delta_1(Y + Z) = \Delta_1 Y + \Delta_1 Z$

If $Y_t = Z_t + \alpha t$ then $(\Delta_1 Y)_t = (\Delta_1 Z)_t + \alpha$: Δ_1 removes linear trends

Repeated application of Δ_1 removes polynomial trends

De-seasonalizing filters

De-seasonalizing R_s: (sum of last s values)

$$R_s Y = X \Leftrightarrow X \text{ has same length as } Y \text{ and}$$
$$X_t = Y_{t-s+1} + \cdots + Y_{t-1} + Y_t$$

with the convention that $Y_t = 0$ whenever $t \leq 0$

If Y_t is periodic of period s then $R_s Y$ is constant

R_s removes periodic components

Differencing Δ_s:

$$\Delta_s Y = X \Leftrightarrow X \text{ has same length as } Y \text{ and } X_t = Y_t - Y_{t-s}$$

with the convention that $Y_t = 0$ whenever $t \leq 0$

De-seasonalizing filters

$$\Delta_s = R_s \Delta_1$$

this means that if $Y \overset{\Delta_1}{\rightarrow} Z \overset{R_s}{\rightarrow} X$ and $Y \overset{\Delta_s}{\rightarrow} X'$ then $X = X'$

Proof:

$$Z_t = Y_t - Y_{t-1}$$
$$X_t = Z_t + \cdots + Z_{t-s+1} = Y_t - Y_{t-1} + Y_{t-1} - Y_{t-2} \ldots + Y_{t-s+1} - Y_{t-s}$$
$$= Y_t - Y_{t-s}$$

Which matrix is the representation of Δ_1 (over time series of length n) ?

A. $A = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ -1 & 1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & 0 & \dots & -1 & 1 \end{pmatrix}$

B. $B = \begin{pmatrix} 1 & -1 & \dots & \dots & \dots & 0 \\ 0 & 1 & -1 & \dots & \dots & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 1 & -1 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$

C. None of these

D. I don't know

Which matrix is the representation of R_4 (over time series of length $n = 6$) ?

A. $A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$

B. $B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$

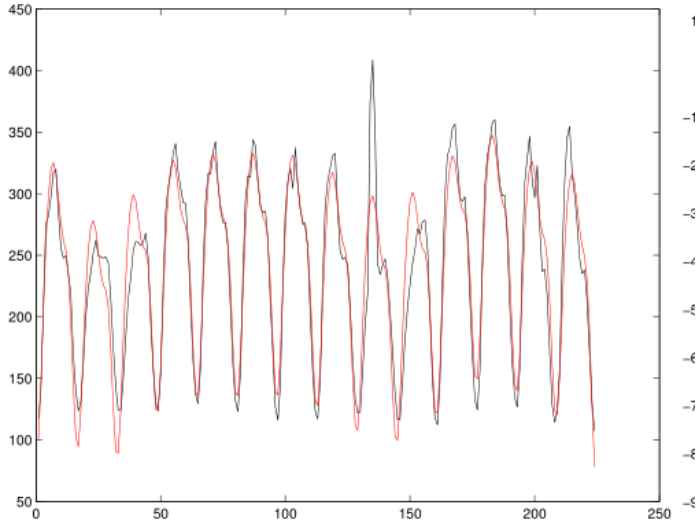
C. None of these

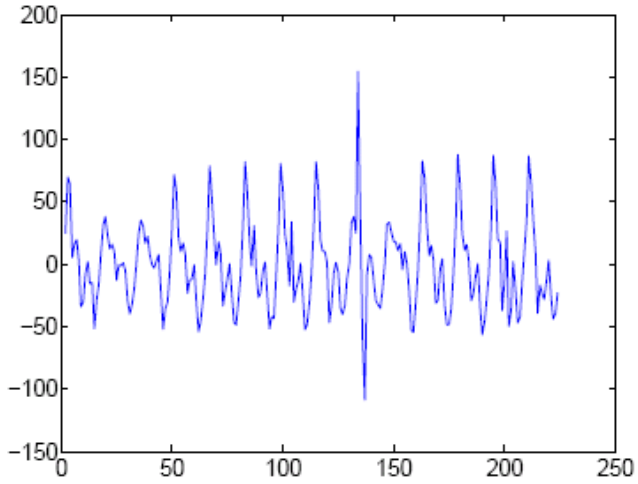
D. I don't know

Say what is true

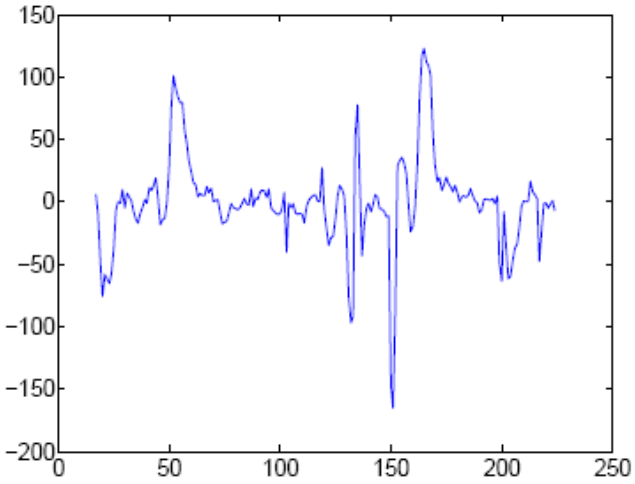
A. A
B. B
C. C
D. A,B
E. A,C
F. B,C
G. All
H. None
I. I don't know

$$A.\ \ R_s\Delta_1 = \Delta_s$$
$$B.\ \ \Delta_1 R_s = \Delta_s$$
$$C.\ \ \Delta_1\Delta_1 = \Delta_2$$

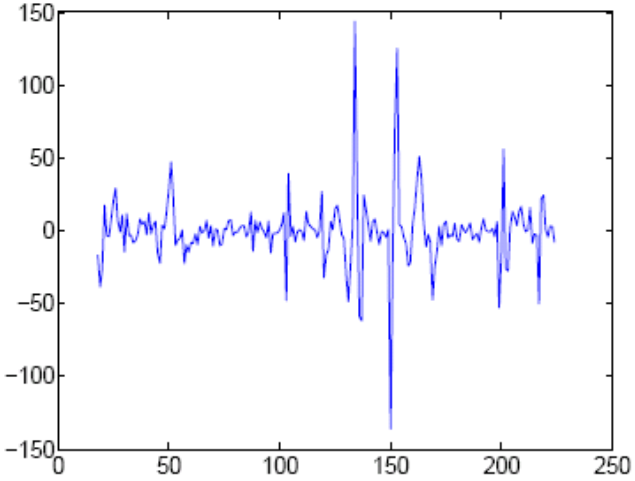




(a) Differencing at Lag 1



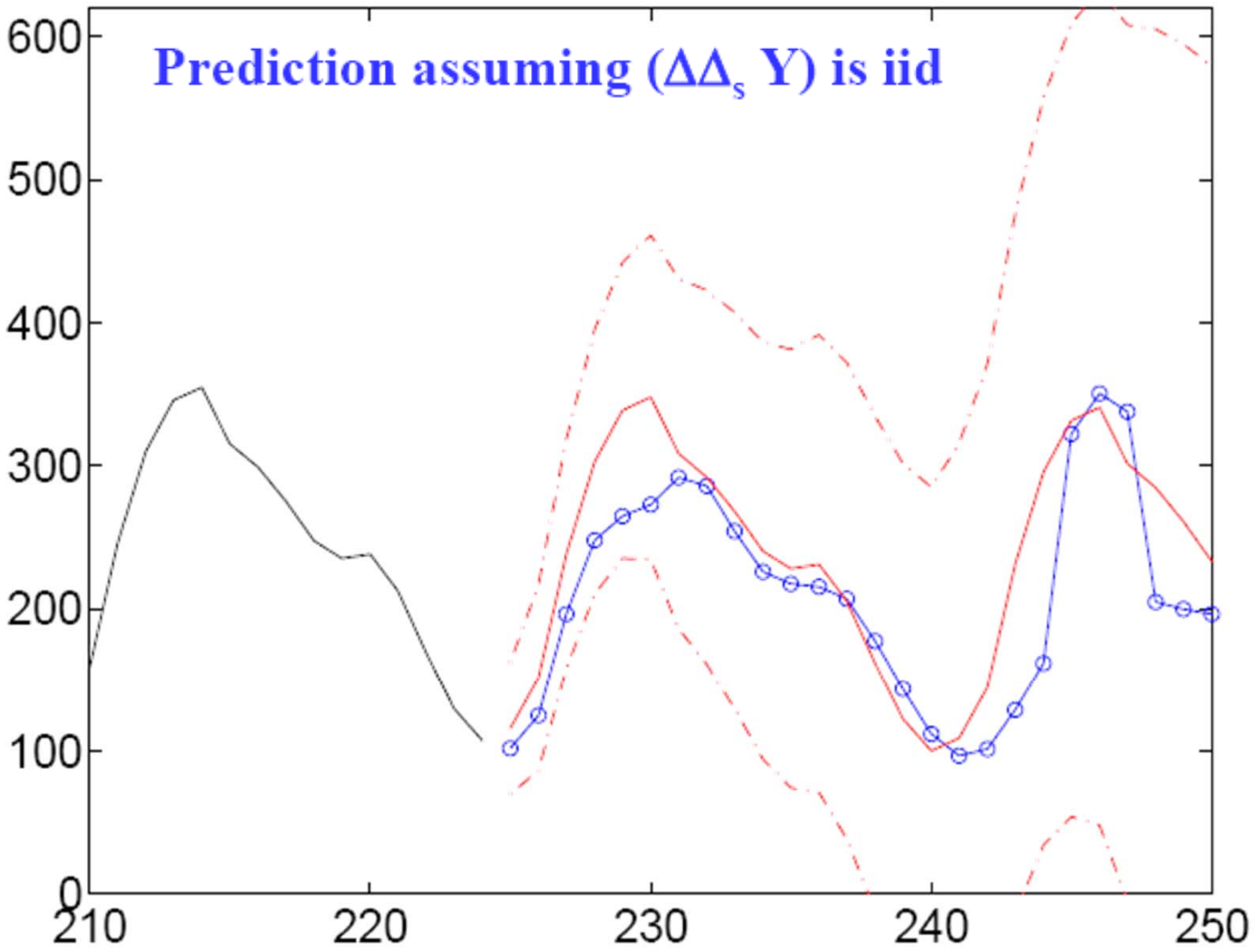
(b) Differencing at Lag 16



(c) Differencing at Lags 1 and 16

EXAMPLE 5.3: INTERNET TRAFFIC. In Figure 5.7 we apply the differencing filter Δ_1 to the time series in Example 5.1 and obtain a strong seasonal component with period $s = 16$. We then apply the de-seasonalizing filter R_{16}; this is the same as applying Δ_{16} to the original data. The result does not appear to be stationary; an additional application of Δ_1 is thus performed.

Point Predictions from Differenced Data



(d) Prediction at time 224

How are these predictions made ? To answer this question,we need to see how to use filters.

7. Filters for Dummies

D.1.1 Backshift Operator

We consider data sequences of finite, but arbitrary length and call $\mathcal{S}$ the set of all such sequences (i.e. $\mathcal{S} = \bigcup_{n=1}^{\infty} \mathbb{R}^n$). We denote with $\text{length}(X)$ the number of elements in the sequence X.

The *backshift* operator is the mapping B from $\mathcal{S}$ to itself defined by:

$$\begin{aligned} \text{length}(BX) &= \text{length}(X) \\ (BX)_1 &= 0 \\ (BX)_t &= X_{t-1} \qquad t = 2, ..., \text{length}(X) \end{aligned}$$

We usually view a sequence $X \in \mathcal{S}$ as a column vector, so that we can write:

$$B \begin{pmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{pmatrix} = \begin{pmatrix} 0 \\ X_1 \\ \dots \\ X_{n-1} \end{pmatrix} \tag{D.1}$$

when $\text{length}(X) = n$.

Backshift Operator in Matrix Form

$$(BX)_t = X_{t-1}$$

B

$$\begin{pmatrix} 0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix}$$

$$(B^2X)_t = X_{t-2}$$

B^2

$$\begin{pmatrix} 0 \\ 0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix}$$

Filters for Dummies

DEFINITION D.1. *A filter (also called "causal filter", or "realizable filter") is any mapping, say F, from $\mathcal{S}$ to itself that has the following properties.*

1. *A sequence of length n is mapped to a sequence of same length.*
2. *There exists an infinite sequence of numbers h_m, $m = 0, 1, 2, ...$ (called the filter's impulse response) such that for any $X \in \mathcal{S}$*

$$(FX)_t = h_0 X_t + h_1 X_{t-1} + ... + h_{t-1} X_1 \qquad t = 1, ..., length(X) \tag{D.4}$$

In matrix form, if we know that length$(X) \leq n$ we can write Eq.(D.4) as

$$FX = \begin{pmatrix} h_0 & 0 & \cdots & 0 & 0 \\ h_1 & h_0 & & \vdots & \vdots \\ h_2 & h_1 & \ddots & & \\ \vdots & \vdots & \ddots & h_0 & 0 \\ h_{n-1} & h_{n-2} & \cdots & h_1 & h_0 \end{pmatrix} X \tag{D.6}$$

Operator Notation

Example: $(FX)_t = 3X_t - 2X_{t-1} + X_{t-2}$

Impulse response: $h_0 = 3, h_1 = -2, h_2 = 1, h_k = 0, k \geq 3$

Matrix form:

$$\begin{pmatrix} 0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ -2 & 3 & 0 & 0 & 0 & 0 \\ 1 & -2 & 3 & 0 & 0 & 0 \\ 0 & 1 & -2 & 3 & 0 & 0 \\ 0 & 0 & 2 & -2 & 3 & 0 \\ 0 & 0 & 0 & 1 & -2 & 3 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{pmatrix}$$

$$M = 3\,\mathrm{Id} - 2\,B + B^2$$

In operator notation we write

$$F = 3 - 2B + B^2$$

where B is the backshift filter.

Filters

Operator notation of F	Input-Output Equation $Y = FX$	Impulse Response
1	$Y_t = X_t$	$(1,0,0,\dots)$
B	$Y_t = X_{t-1}$	$(0,1,0,0,\dots)$
$\Delta_1 = 1 - B$	$Y_t = X_t - X_{t-1}$	$(1,-1,0,0,\dots)$
$\Delta_s = 1 - B^s$	$Y_t = X_t - X_{t-s}$	$(1,0,\dots,0,-1,0,0,\dots)$
$R_s = 1 + B + \cdots + B^{s-1}$	$Y_t = X_t + \cdots + X_{t-s+1}$	$(1,\dots,1,0,0,\dots)$
$F = h_0 + h_1 B + h_2 B^2 + \cdots$	$Y_t = h_0 X_t + h_1 X_{t-1} + \cdots$	$(h_0, h_1, \dots)$
$\frac{1}{F}$ defined when $h_0 \neq 0$	$X_t = h_0 Y_t + h_1 Y_{t-1} + \cdots$ i.e. $Y_t = \frac{1}{h_0} X_t - \frac{h_1}{h_0} Y_{t-1} - \cdots$	

Impulse Response

Let F be a filter with impulse response $h_0, h_1, \ldots$

Thus $Y_t = h_0 X_t + h_1 X_{t-1} \ldots$

If the input X is the *impulse*

$$X = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \ldots \end{pmatrix}$$

then the output it

$$Y = \begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ \ldots \end{pmatrix}$$

Inverse of a Filter

$$FX = \begin{pmatrix} h_0 & 0 & \cdots & 0 & 0 \\ h_1 & h_0 & & \vdots & \vdots \\ h_2 & h_1 & \ddots & & \\ \vdots & \vdots & \ddots & h_0 & 0 \\ h_{n-1} & h_{n-2} & \cdots & h_1 & h_0 \end{pmatrix} X$$

Filter F has an inverse if and only if $h_0 \neq 0$

Example: $\Delta_1 = 1 - B$ is invertible

Calculus of Filters

FG means the operator composition (G followed by F):

$$Y \xrightarrow{G} Z \xrightarrow{F} X \text{ implies } Y \xrightarrow{FG} X$$

Filters commute: $FG = GF$ Magical !

$\frac{1}{F}$ means F^{-1}, the inverse of F, defined if $h_0 \neq 0$

$$\frac{G}{F} = G\,F^{-1} = F^{-1}G$$

$F + G$ means the algebraic sum: $[(F+G)X]_t = [FX]_t + [GX]_t$

Let $\boldsymbol{F}$ be the filter defined by $\boldsymbol{Y = FX}$ with

$$\boldsymbol{Y_t = X_t - 3X_{t-1} + 2X_{t-2}}$$

Say what is true

A. $F = 1 - 3B + 2B^2$

B. The impulse response of F is $(1, -3, 2, 0, 0 \ldots)$

C. A and B

D. None

E. I don't know

Let $\boldsymbol{F}$ be the filter defined by $\boldsymbol{Y = FX}$ with

$$\boldsymbol{Y_t - 3Y_{t-1} + 2Y_{t-2} = X_t}$$

Say what is true

A. $F = \frac{1}{1-3B+2B^2}$

B. The impulse response of F is $(1, -3, 2, 0, 0 \ldots)$

C. A and B

D. None

E. I don't know

ARMA Filters and matlab's filter function

Finite Impulse Response (FIR) also called Moving Average (MA) filter: $h_k = 0$ for k large enough $\Leftrightarrow$ F is polynomial in B

Example: $\Delta_1 = 1 - B$ is FIR; $\Delta_1^{-1} = 1 + B + B^2 + \cdots$ is not FIR

$F = h_0 + h_1 B + \cdots + h_p B^p$ is the generic FIR filter

Auto-Regressive (AR) Filter is the inverse of a FIR filter

$1 + B + B^2 + \cdots = \frac{1}{1-B}$ is AR filter

$F = \frac{1}{h_0 + h_1 B + \cdots + h_p B^p}$ with $h_0 \neq 0$ is the generic FIR filter

ARMA Filter is F/G where F, G are FIR

$F = \frac{f_0 + f_1 B + \cdots + f_p B^p}{1 + g_1 B + \cdots + g_q B^q}$ is the generic ARMA filter

Implemented by Matlab's filter() function

- `Y = filter(P,Q,X)` computes the output $Y = [y_1\ y_2\ y_3 ... y_n]$ of the filter, where $P = [P_0\ P_1\ P_2 ... P_p]$, $Q = [1\ Q_1\ Q_2 ... Q_q]$ are the filter coefficients and $X = [x_1\ x_2\ x_3 ...]$ is the input. The filter is defined by the relation

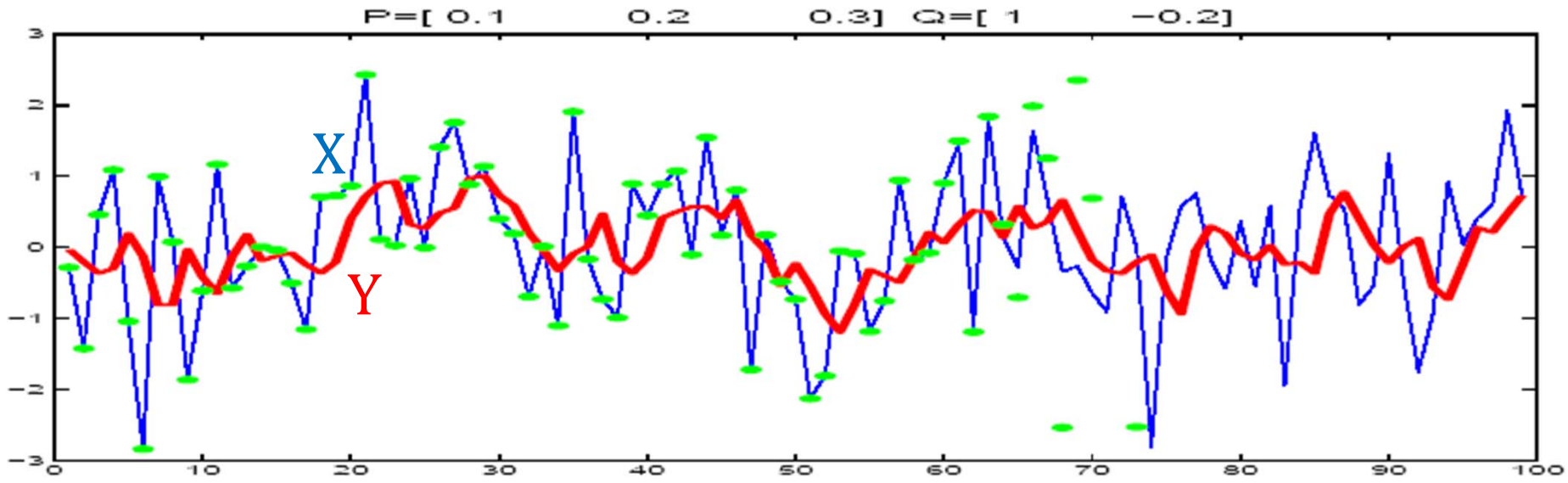
$$y_k + Q_1 y_{k-1} + ... + Q_q y_{k-p} = P_0 x_k + P_1 x_{k-1} + ... + P_q x_{k-q}$$

where we set $x_i = 0$ and $y_i = 0$ when $i < 0$ or $i > n$.
The polynomial $P(\xi) = P_0 \xi^p + P_1 \xi^{q-1} + ... + P_q$ is called the *numerator polynomial* and $Q(\xi) = \xi^q + Q_1 \xi^{q-1} + ... + Q_q$ the *denominator polynomial*.
In our terminology, this filter is the mapping

$$\begin{array}{rcl} \mathbb{R}^n & \rightarrow & \mathbb{R}^n \\ X & \rightarrow & Y = \dfrac{\sum_{i=0}^{p} P_p B^p}{Id + \sum_{j=1}^{q} Q_j B}(X) = \dfrac{P(B)}{Q(B)} \cdot X \end{array}$$

Matlab	Operator Notation	Input-Output Equation
Y= filter ([0.1 0.2 0.3], [1 -0.2], X)	$Y = \frac{0.1+0.2B+0.3B^2}{1-0.2B}\ X$	$Y_t - 0.2Y_{t-1} = 0.1X_t + 0.2X_{t-1} + 0.3\ X_{t-2}$

A sample of $Y = \frac{0.1+0.2B+0.3B^2}{1-0.2B} X$



Q: how can we compute X back from Y ?

A: inverse the filter $X = \frac{1-0.2B}{0.1+0.2B+0.3B^2} Y$

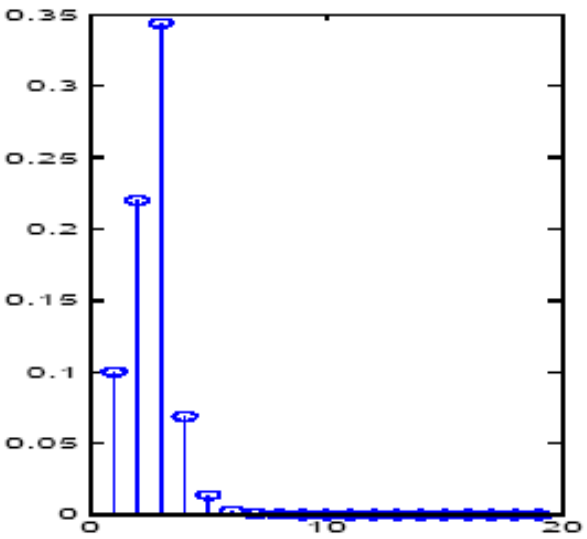
The inverse of $Y = \text{filter}(P, Q, X)$ is $X = \text{filter}(Q, P, X)$ defined if first element of Q is $\neq 0$

The result is shown with green dots; after $t = 60$ the results are incorrect. Why ?

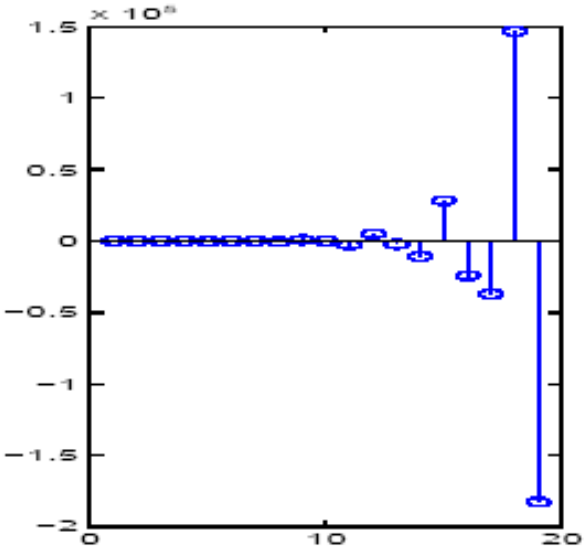
To understand what happens, let us compute the coefficients of these filters (i.e. their impulse responses)

It is obtained by $h = \mathrm{filter}(P, Q, \mathrm{imp})$ where $\mathrm{imp} = [1\ 0\ 0\ \ldots]$ is called an impulse

Impulse response of $F = \frac{0.1+0.2B+0.3B^2}{1-0.2B}$



Impulse response of F^{-1}



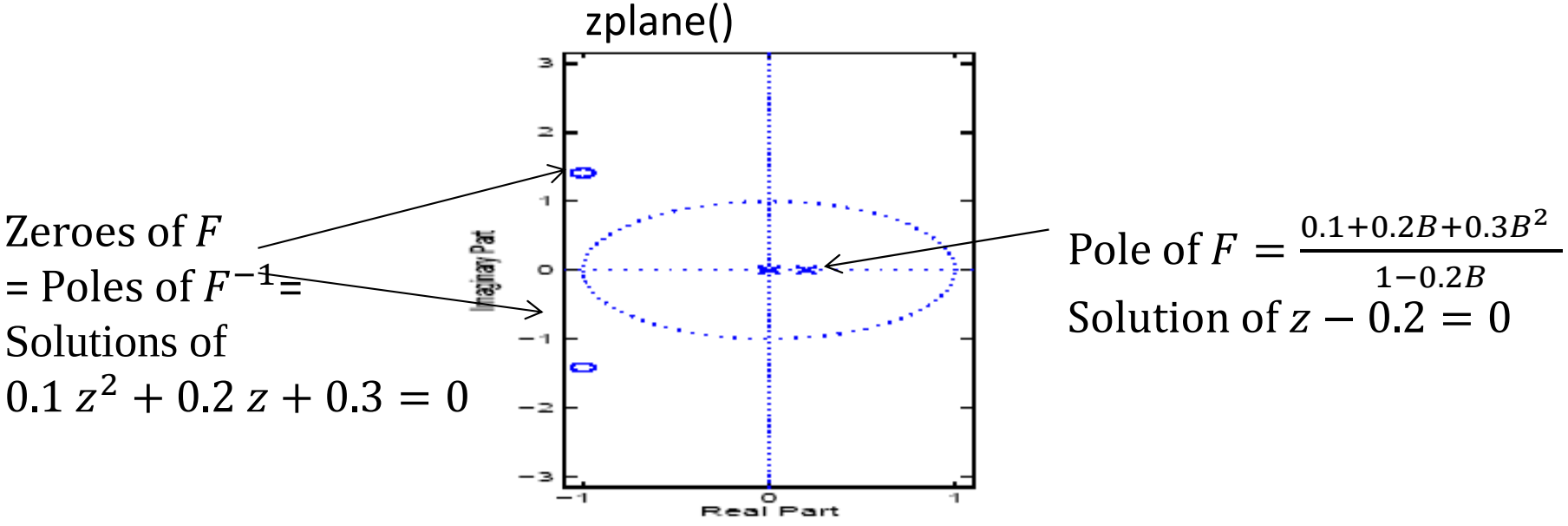
The impulse of F^{-1} grows exponentially and becomes huge → numerical computation becomes impossible

Filter BIBO Stability: $\sum_n |h_n| < \infty$

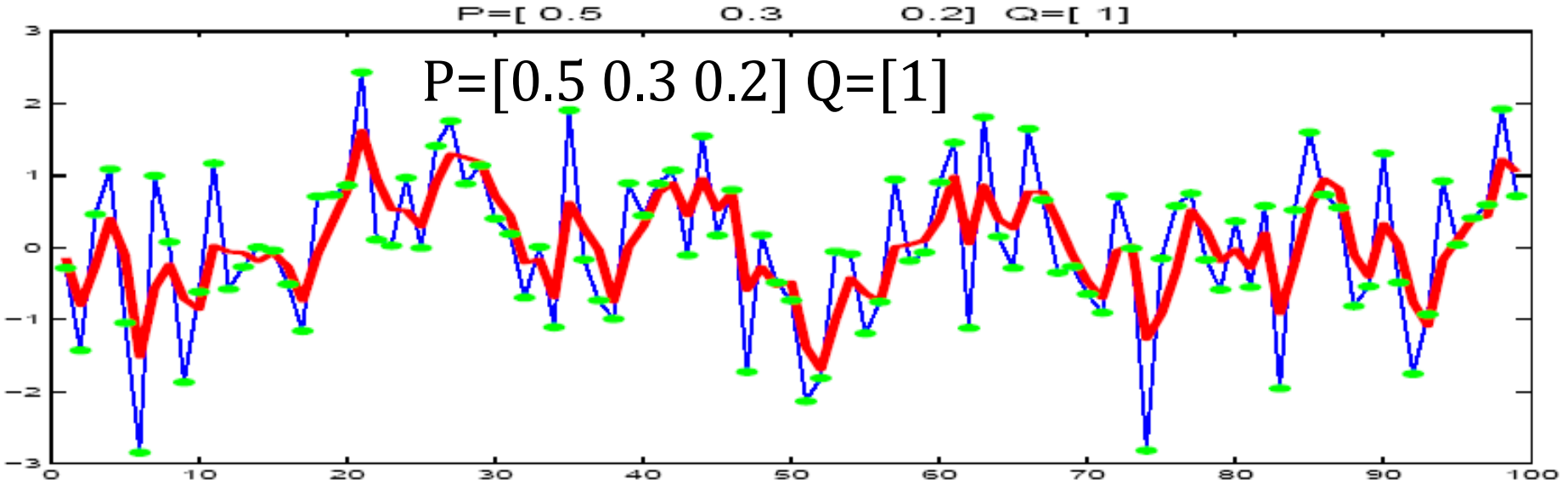
A filter that is unstable usually causes numerical problems (accumulation of rounding errors)

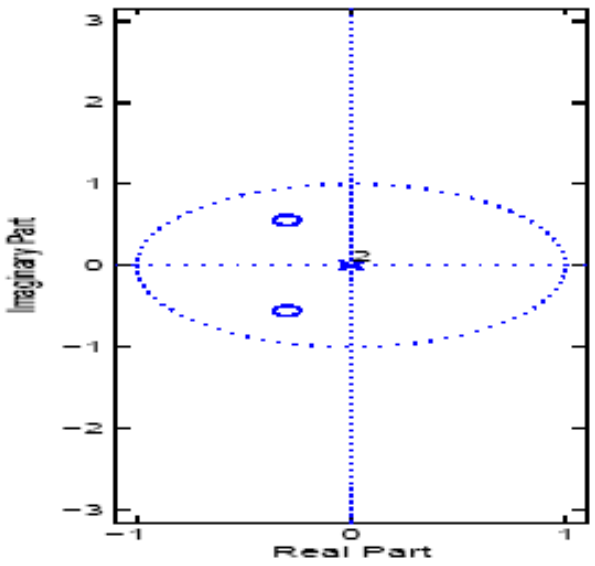
For an ARMA filter $F = \frac{a_0 + a_1 B + \cdots + a_p B^p}{b_0 + b_1 B + \cdots + b_q B^q}$, the poles are the q (complex) roots of the denominator polynomial $Q(\xi) = b_0 \xi^q + \cdots + b_q$

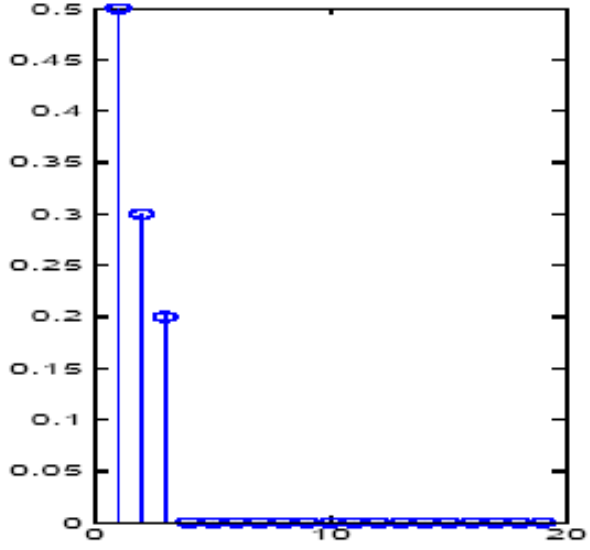
Stable $\Leftrightarrow$ [$q = 0$ (no pole) or all poles have module < 1]

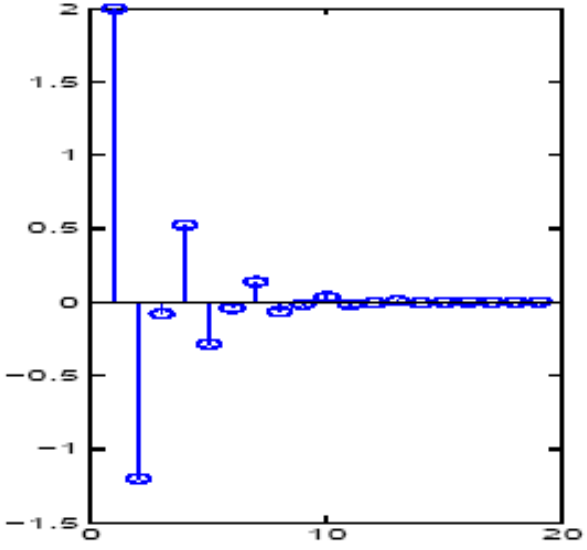


A filter with stable inverse

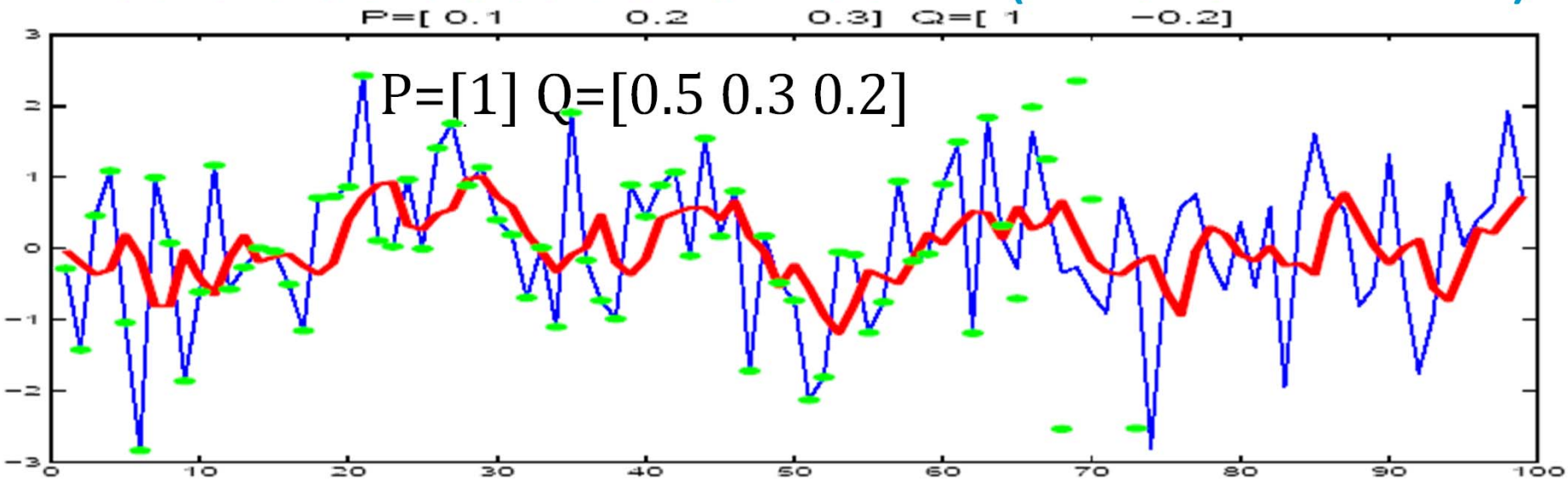








What is true about this filter $\boldsymbol{F}$ (where $Y = FX$)?



A. $0.5Y_t + 0.3\,Y_{t-1} + 0.2Y_{t-2} = X_t$

B. $Y_t = X_t - 0.5Y_t - 0.3\,Y_{t-1} - 0.2Y_{t-2}$

C. $Y_t = \dfrac{X_t}{0.5Y_t + 0.3Y_{t-1} + 0.2\,Y_{t-2}}$

D. A and B

E. A and C

F. B and C

G. All

H. None

I. I don't know

MA(∞) and AR(∞) representation of a filter F

Let $Y = FX$ for a filter F with impulse response and h_i

The standard input-output equation

$$Y_t = h_0 X_t + h_1 X_{t-1} + \cdots + h_{t-1} X_1$$

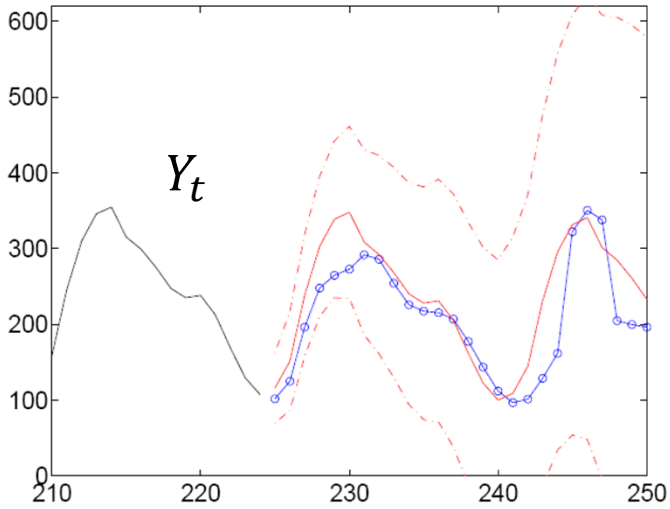
is called MA (∞) representation F.

If F is invertible, let h_i' be the impulse response of F^{-1} so that $X_t = h_0' Y_t + h_1' Y_{t-1} + \cdots h_{t-1}' Y_1$ and thus

$$Y_t = \frac{1}{h_0'} X_t - \frac{h_1'}{h_0'} Y_{t-1} - \frac{h_2'}{h_0'} Y_{t-2} - \cdots - \frac{h_{t-1}'}{h_0'} Y_1$$

This is called the AR (∞) representation of F.

8. How is this prediction done ?



(d) Prediction at time 224

Recall that $X = LY$ with

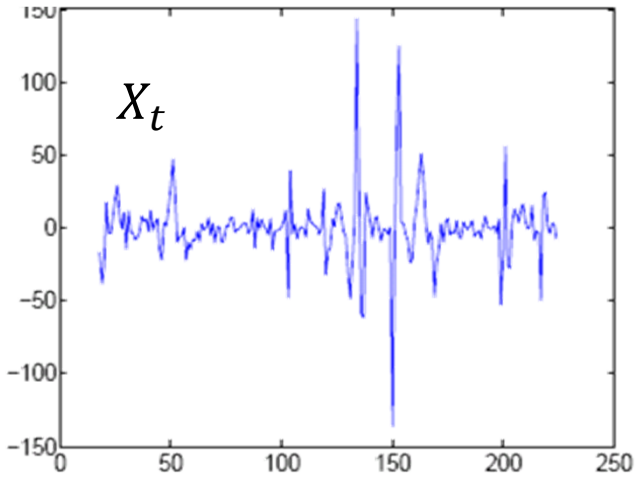
$L = \Delta_1\Delta_{16}$ and we assume $X \sim \text{iid } F()$

thus

$$X_t = Y_t - Y_{t-1} - Y_{t-16} + Y_{t-17}$$

$$Y_t = X_t + Y_{t-1} + Y_{t-16} - Y_{t-17}$$

(MA representation of L)



(c) Differencing at Lags 1 and 16

Prediction at lag $\ell = 1$:

assume we know $Y_1, \ldots, Y_t$

$$Y_{t+1} = X_{t+1} + Y_t + Y_{t-15} - Y_{t-16}$$

known

Given the past up to time t, this is random with distribution $F()$

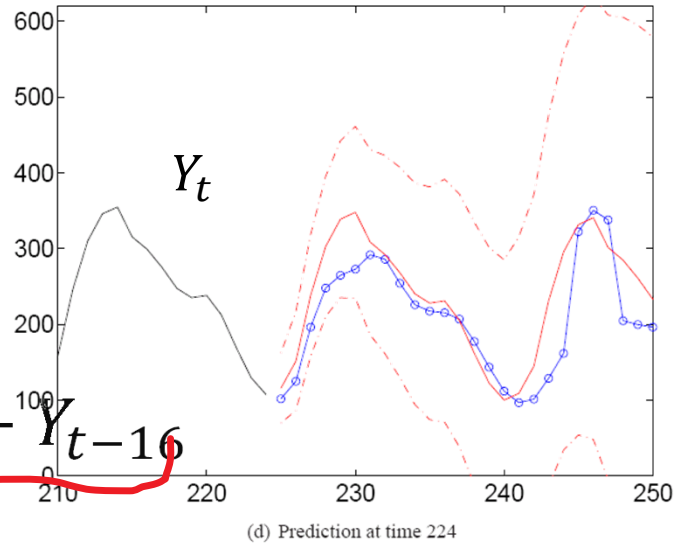
Point Prediction at lag 1

Prediction at lag $\ell = 1$:
assume we know $Y_1, \ldots, Y_t$

$$Y_{t+1} = X_{t+1} + Y_t + Y_{t-15} - Y_{t-16}$$

known

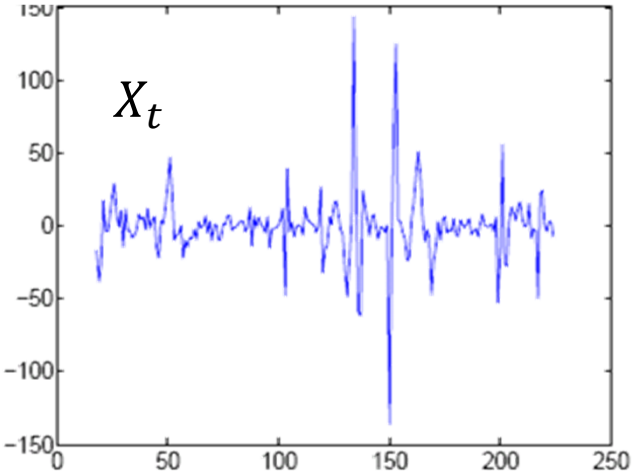
Given the past up to time t, this is random with distribution $F()$



(d) Prediction at time 224

Assume $X \sim iid\ F()$ with zero mean,
the mean of Y_{t+1} given the past up to time t is (point prediction)

$$\hat{Y}_t(1) = Y_t + Y_{t-15} - Y_{t-16}$$



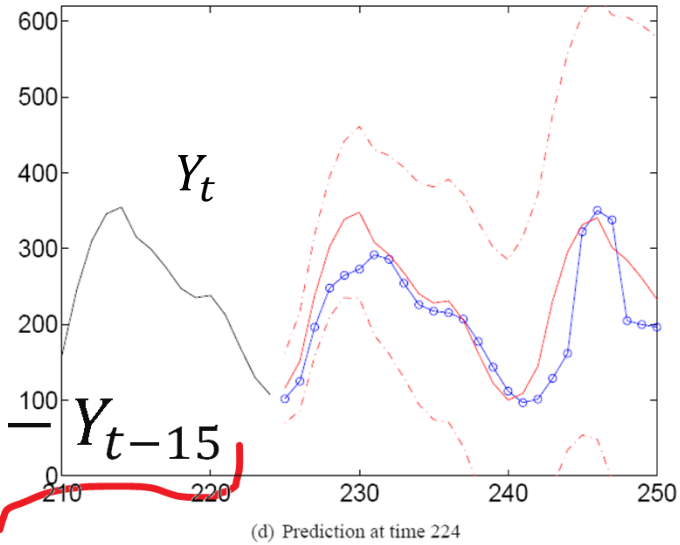
(c) Differencing at Lags 1 and 16

Point Predictions

Prediction at lag $\ell = 2$:

assume we know $Y_1, \ldots, Y_t$

$$Y_{t+2} = X_{t+2} + Y_{t+1} + Y_{t-14} - Y_{t-15}$$

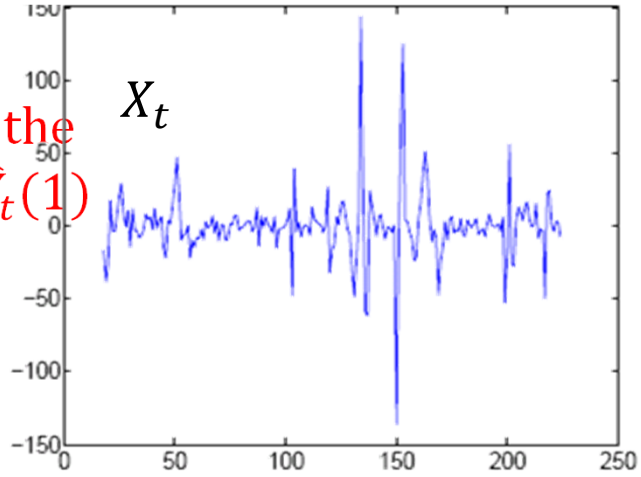


(d) Prediction at time 224

known

Given the past up to time t, the conditional expectation is 0 (F() has zero mean)

Given the past up to time t, the conditional expectation is $\hat{Y}_t(1)$



(c) Differencing at Lags 1 and 16

Therefore : (point prediction at lag 2)

$$\hat{Y}_t(2) = \hat{Y}_t(1) + Y_{t-14} - Y_{t-15}$$

At lag ℓ: use the formula

$$Y_{t+\ell} = X_{t+\ell} + Y_{t+\ell-1} + Y_{t+\ell-16} - Y_{t+\ell-17}$$

in which you replace

Y_{t+s} by $\hat{Y}_t(s)$ for $s > 0$ and $X_{t+\ell}$ by 0 (= the mean of $F()$)

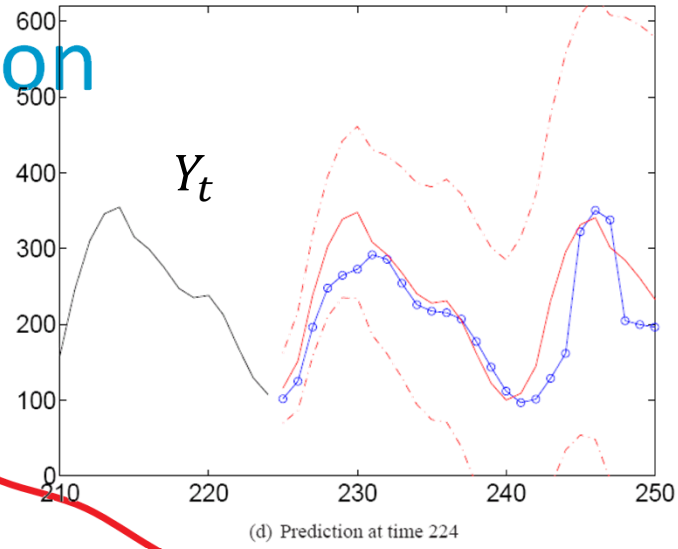
for example $\hat{Y}_t(17) = \hat{Y}_t(16) + \hat{Y}_t(1) - Y_t$

PROPOSITION 5.1. *Assume that $X = LY$ where L is a differencing or de-seasonalizing filter with impulse response $g_0 = 1, g_1, ..., g_q$. Assume that we are able to produce a point prediction $\hat{X}_t(\ell)$ for $X_{t+\ell}$ given that we have observed X_1 to X_t. For example, if the differenced data can be assumed to be iid with mean μ, then $\hat{X}_t(\ell) = \mu$.*

A point prediction for $Y_{t+\ell}$ can be obtained iteratively by:

$$\begin{aligned} \hat{Y}_t(\ell) &= \hat{X}_t(\ell) - g_1\hat{Y}_t(\ell-1) - \ldots - g_{\ell-1}\hat{Y}_t(1) - g_\ell y_t - \ldots \\ &\quad -g_q y_{t-q+\ell} \quad \text{for } 1 \le \ell \le q \qquad (5.14) \\ \hat{Y}_t(\ell) &= \hat{X}_t(\ell) - g_1\hat{Y}_t(\ell-1) - \ldots - g_q\hat{Y}_t(\ell-q) \text{ for } \ell > q \qquad (5.15) \end{aligned}$$

Use of the alternative representation (MA representation of L^{-1})



(d) Prediction at time 224

Prediction at lag $\ell = 1$:

assume we know $Y_1, \ldots, Y_t$

$$Y_{t+1} = X_{t+1} + X_t + \cdots + X_{t-14}$$
$$+2(X_{t-15} + X_{t-16} + \cdots + X_{t-30})$$
$$+3(X_{t-31} + X_{t-32} + \cdots + X_{t-46}) + \cdots$$

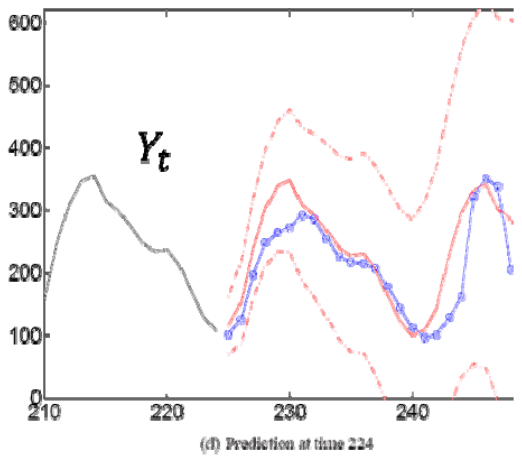
known

Given the past up to time t, this is random with distribution $F()$

therefore $\hat{Y}_t(1) = X_t + \cdots + X_{t-14} + 2(X_{t-15} + X_{t-16} + \cdots + X_{t-30}) + 3(X_{t-31} + X_{t-32} + \cdots + X_{t-46}) + \cdots$

Note it would not be a good idea to use this formula to compute $\hat{Y}_t(1)$ because we accumulate a large number of errors – but it can be used to compute prediction intervals

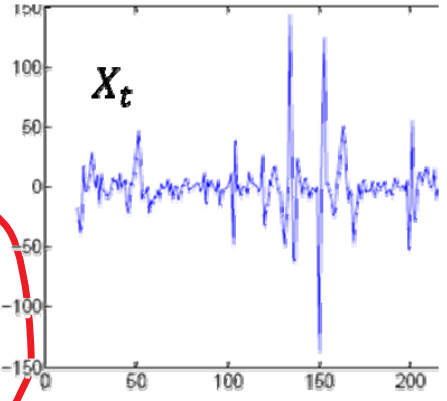
Computation of Prediction Intervals (example with $\ell = 3$)



(d) Prediction at time 224

Prediction at lag $\ell = 3$: assume we know $Y_1, \ldots, Y_t$

Since the filter L is causal and invertible, knowing $Y_1, \ldots, Y_t$ is equivalent to knowing $X_1, \ldots, X_t$



(c) Differencing at Lags 1 and 1

$$Y_{t+3} = X_{t+3} + X_{t+2} + X_{t+1} + X_t + \cdots + X_{t-12}$$
$$+2(X_{t-13} + X_{t-14} + \cdots + X_{t-28})$$
$$+3(X_{t-39} + X_{t-30} + \cdots + X_{t-44}) + \cdots$$

Known at time t

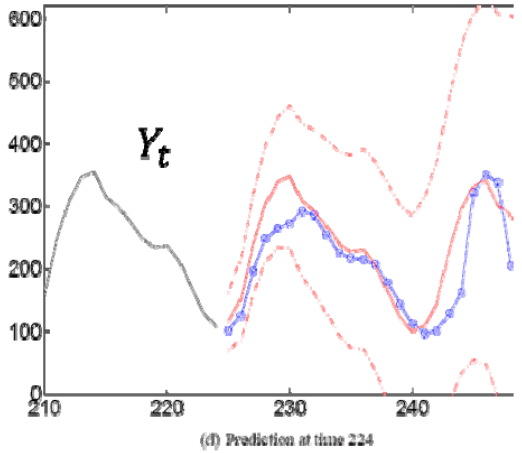
$= \hat{Y}_t(3)$

Therefore

(innovation formula):

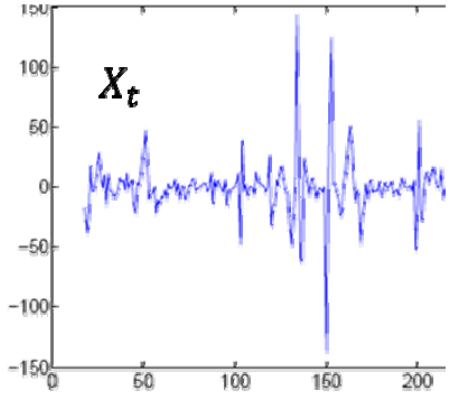
$$Y_{t+3} = X_{t+3} + X_{t+2} + X_{t+1} + \hat{Y}_t(3)$$

$$Y_{t+3} = X_{t+3} + X_{t+2} + X_{t+1} + \hat{Y}_t(3)$$



(d) Prediction at time 224

Given the past up to time t, the distribution of Y_{t+3} is given by
- a constant $\hat{Y}_t(3)$
- plus the sum of 3 independent random variables each with distribution $F()$ (the assumed distribution of X_t)



(e) Differencing at Lags 1 and 1

Example: assume $X_t \sim iid\ N(0, \sigma^2)$
the distribution of $X_{t+3} + X_{t+2} + X_{t+1}$ is $N(0, 3\sigma^2)$

i.e. the distribution of Y_{t+3} given the past up to time t is normal with mean $\hat{Y}_t(3)$ and variance $3\,\sigma^2$

A 95%-prediction interval at lag 3 is…

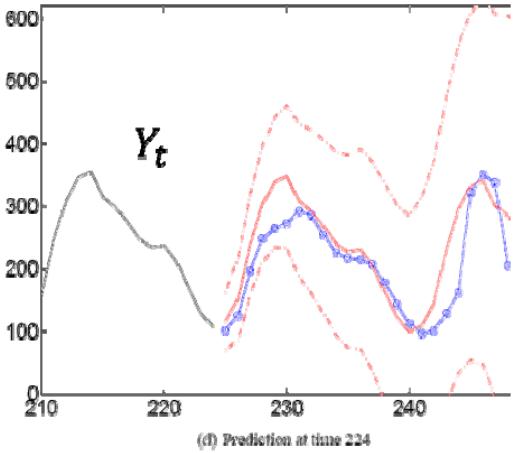
A. $Y_t(3) \pm 1.96\,\sigma$

B. $\hat{Y}_t(3) \pm 1.96 \times \sqrt{3}\,\sigma$

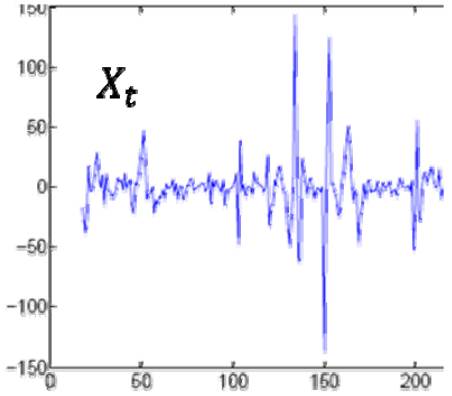
C. $\hat{Y}_t(3) \pm 1.96 \times 3\,\sigma$

D. $\hat{Y}_t(3) \pm 1.96 \times 3\,\frac{\sigma}{\sqrt{n}}$

E. None of the above

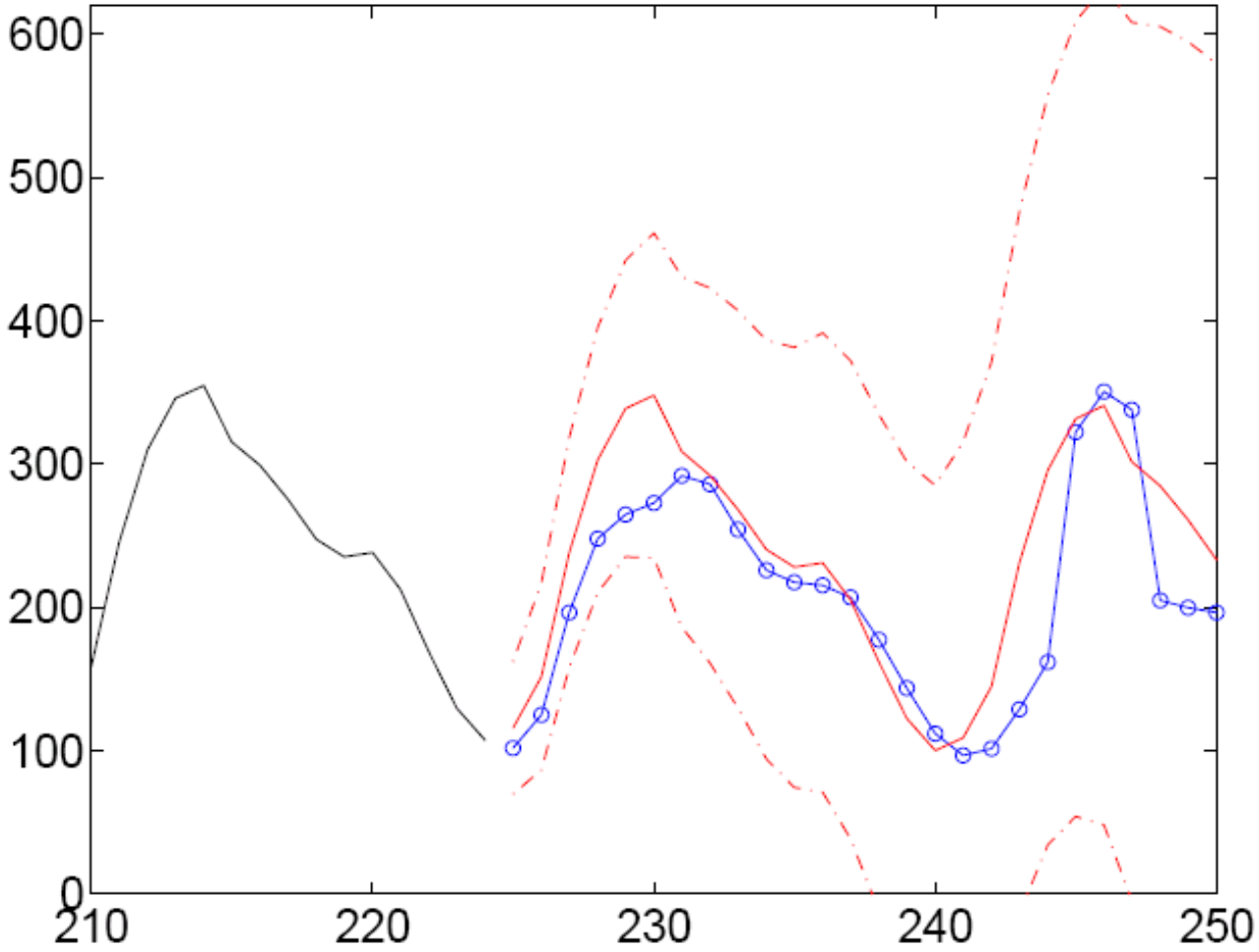


(d) Prediction at time 224



(e) Differencing at Lags 1 and 1

Prediction assuming differenced data is iid $N(0, \sigma^2)$



(d) Prediction at time 224

Figure 6.7: Differencing filters Δ_1 and Δ_{16} applied to Example 6.1 (first terms removed). The forecasts are made assuming the differenced data is iid gaussian with 0 mean. o = actual value of the future (not used for fitting the model).

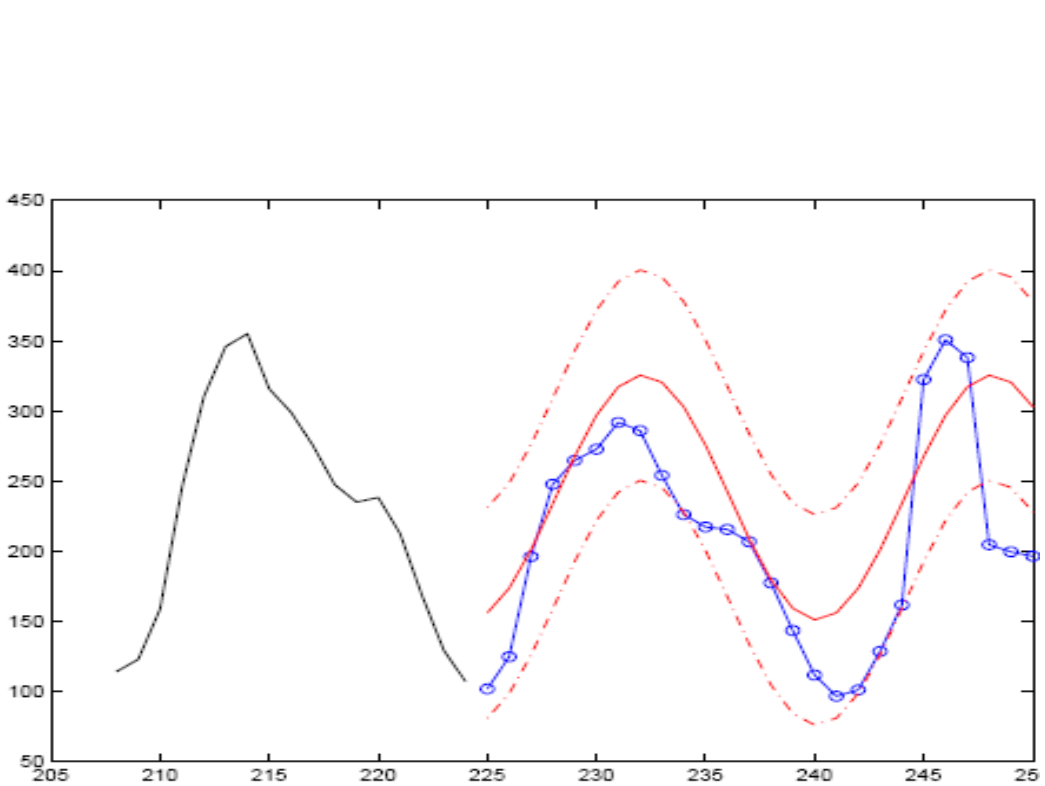
PROPOSITION 5.2. *Assume that the differenced data is iid gaussian. i.e.* $X_t = (LY)_t \sim$ *iid* $N(\mu, \sigma^2)$. *The conditional distribution of* $Y_{t+\ell}$ *given that* $Y_1 = y_1, ..., Y_t = y_t$ *is gaussian with mean* $\hat{Y}_t(\ell)$ *obtained from Eq.(5.14) and variance*

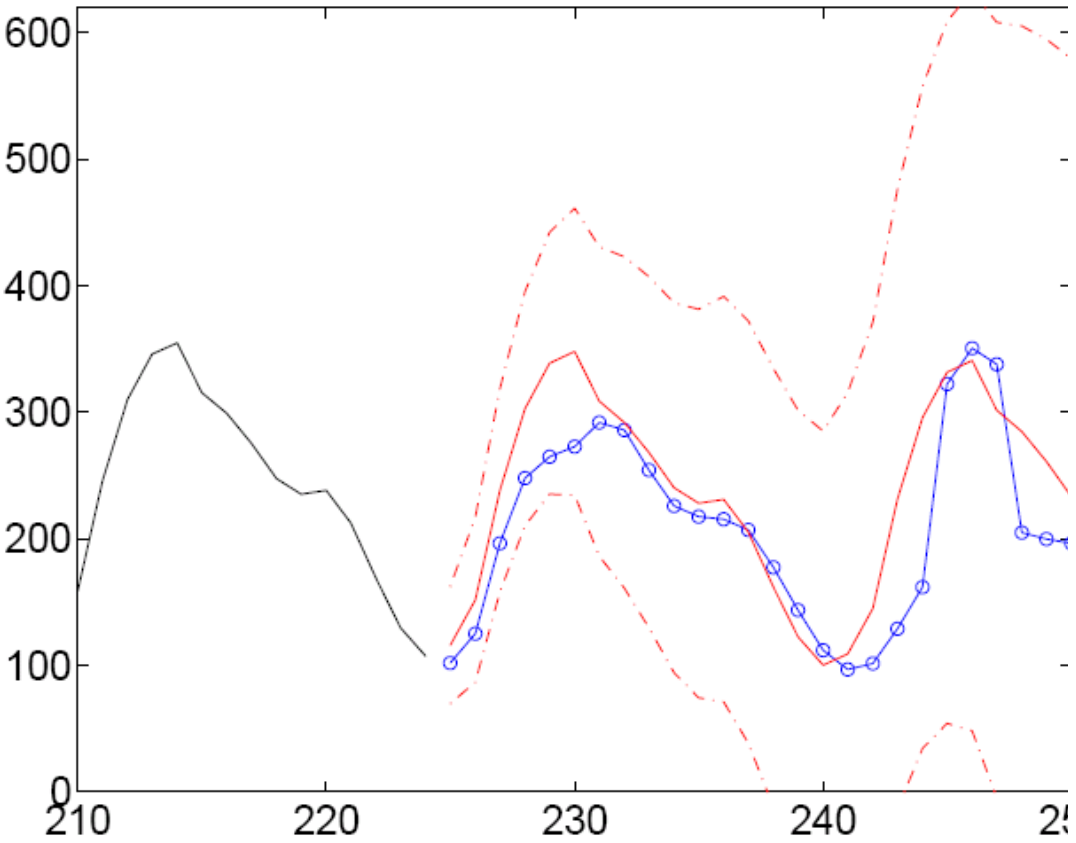
$$MSE_t^2(\ell) = \sigma^2 \left(h_0^2 + \cdots + h_{\ell-1}^2\right) \tag{5.16}$$

where $h_0, h_1, h_2, \ldots$ *is the impulse response of* L^{-1}. *A prediction interval at level* 0.95 *is thus*

$$\hat{Y}_t(\ell) \pm 1.96\sqrt{MSE_t^2(\ell)} \tag{5.17}$$

Compare the Two





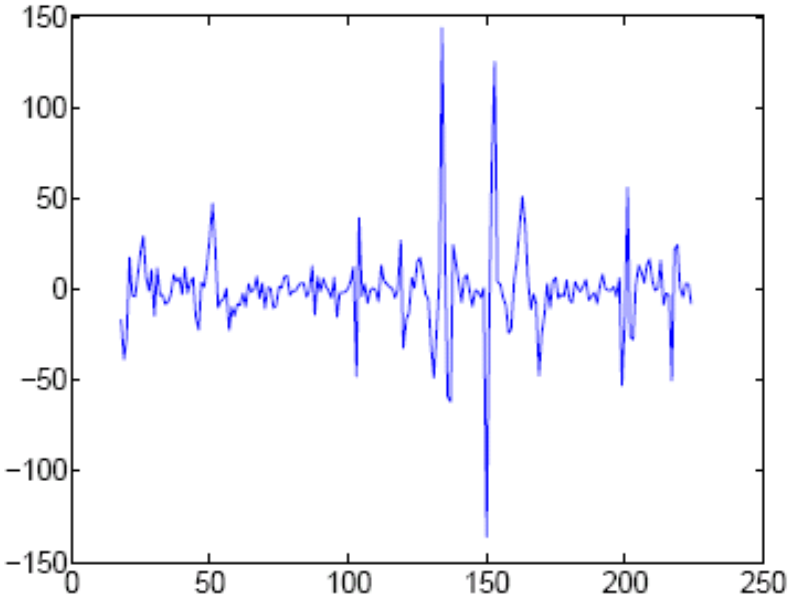
Linear Regression with 3 parameters + variance

Assuming differenced data is iid

9. Using ARMA Models for the Noise

This technique is used when the differenced data appears stationary but not iid – the correlation structure can be used to gain some information about futures

The differenced data can be modelled as an ARMA process instead of iid



(c) Differencing at Lags 1 and 16

Deciding whether a stationary X_t is iid

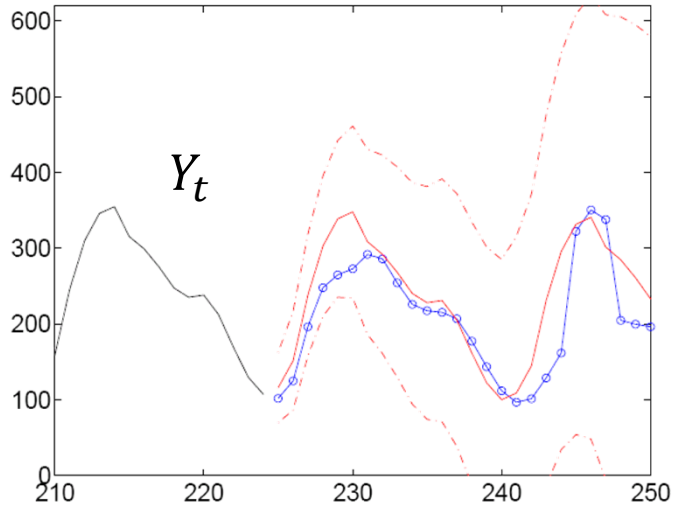
Assume differenced time series X_t looks stationary

Sample auto-covariance $\hat{\gamma}_t = \sum_{s=1}^{n-t}(X_{t+s} - \bar{X})(X_s - \bar{X})$

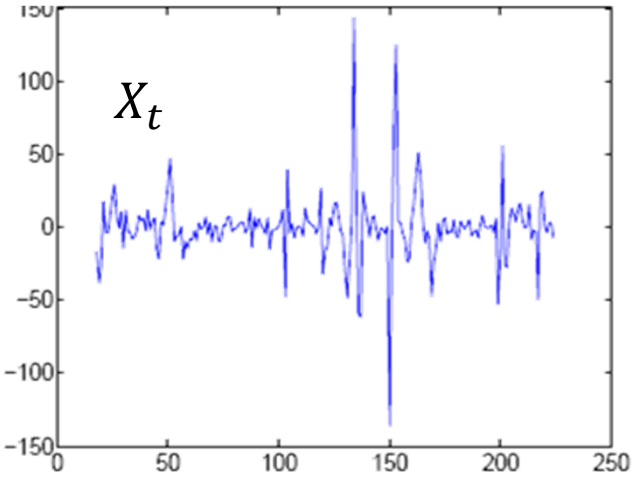
Sample Auto-Correlation Function (ACF) $\rho_t = \frac{\hat{\gamma}_t}{\hat{\gamma}_0}$

If n is large and X_t is iid, around 95% of the the values of ACF lie within $\pm\frac{1.96}{\sqrt{n}}$

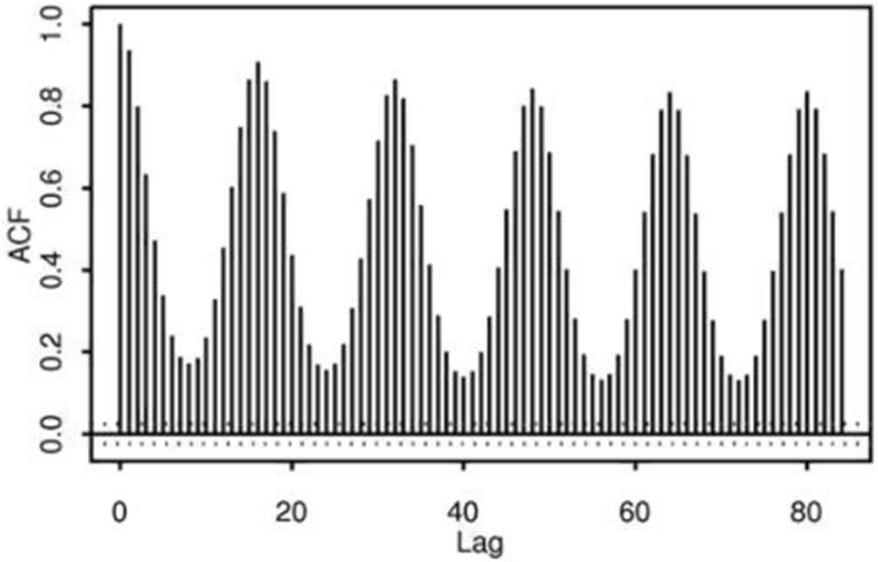
See also tests (Ljung-Box test)



(d) Prediction at time 224



(c) Differencing at Lags 1 and 16



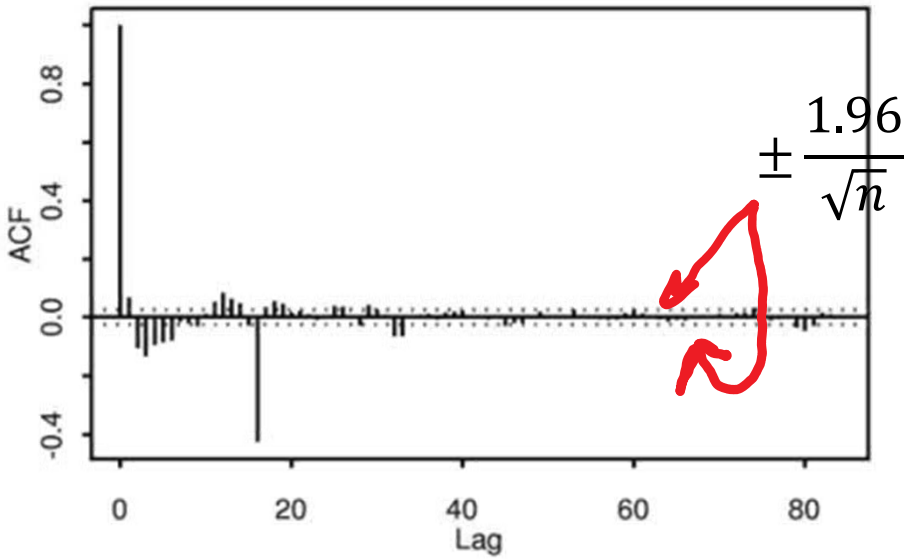


Figure 5.10: First panel: Sample ACF of the internet traffic of Figure 5.1. The data does not appear to come from a stationary process so the sample ACF cannot be interpreted as estimation of a true ACF (which does not exist). Second panel: sample ACF of data differenced at lags 1 and 16. The sampled data appears to be stationary and the sample ACF decays fast. The differenced data appears to be suitable for modelling by an ARMA process.

ARMA Process

DEFINITION 5.1. *A 0-mean ARMA(p, q) process X_t is a process that satisfies for $t = 1, 2, \cdots$ a difference equation such as:*

$$X_t + A_1 X_{t-1} + \cdots + A_p X_{t-p} = \epsilon_t + C_1 \epsilon_{t-1} + \cdots + C_q \epsilon_{t-q} \qquad \epsilon_t \text{ iid } \sim N_{0,\sigma^2} \tag{5.21}$$

Unless otherwise specified, we assume $X_{-p+1} = \cdots = X_0 = 0$.

An ARMA(p, q) process with mean μ is a process X_t such that $X_t - \mu$ is a 0 mean ARMA process and, unless otherwise specified, $X_{-p+1} = \cdots = X_0 = \mu$.

The parameters of the process are $A_1, \cdots, A_p$ (*auto-regressive coefficients*), $C_1, \cdots, C_q$ (*moving average coefficients*) and σ^2 (*white noise variance*). The iid sequence ϵ_t is called the noise sequence, or *innovation*.

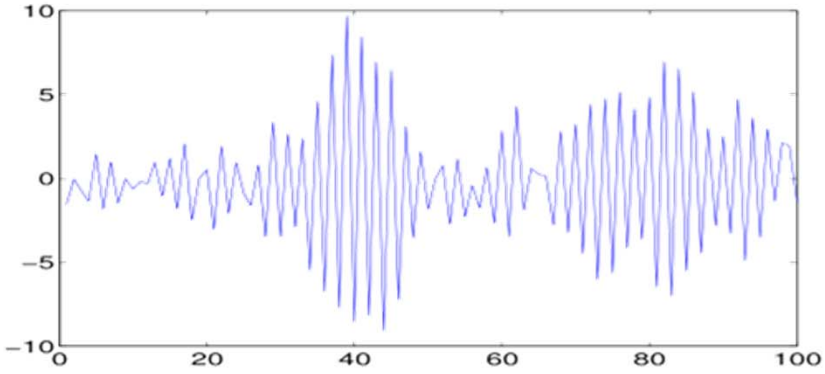
An ARMA$(p, 0)$ process is also called an *Auto-regressive* process, AR(p); an ARMA$(0, q)$ process is also called a *Moving Average* process, MA(q).

$$X = \mu + F\epsilon$$

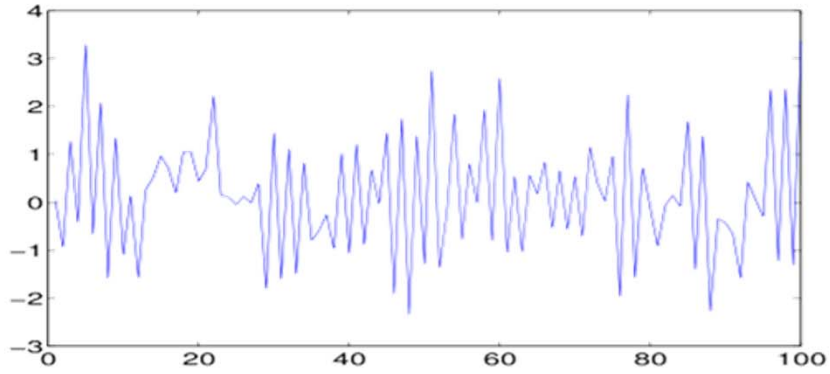
$$F = \frac{1 + C_1 B + \ldots + C_q B^q}{1 + A_1 B + \ldots + A_p B^p}$$

HYPOTHESIS 5.1. *The filter in Eq.(5.23) and its inverse are stable.*

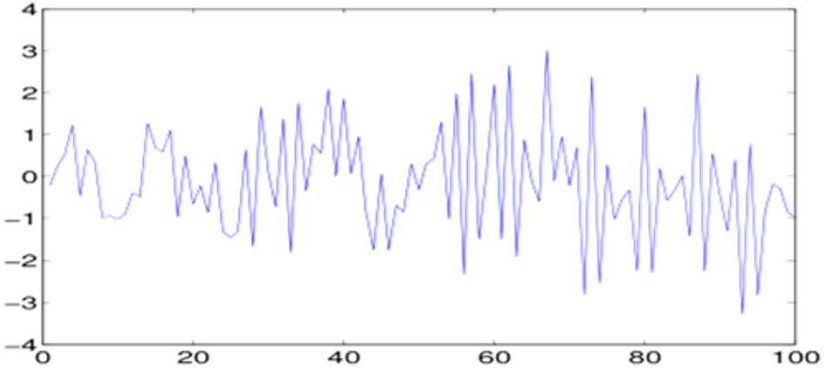
In practice, this means that the zeroes of $1 + A_1 z^{-1} + \ldots + A_p z^{-p}$ and of $1 + C_1 z^{-1} + \ldots + C_q z^{-q}$ are within the unit disk.



(a) ARMA(2,2) $X_t = -0.4X_{t-1} + 0.45X_{t-2} + \epsilon_t - 0.4\epsilon_{t-1} + 0.95\epsilon_{t-2}$



(b) AR(2) $X_t = -0.4X_{t-1} + 0.45X_{t-2} + \epsilon_t$



(c) MA(2) $X_t = \epsilon_t - 0.4\epsilon_{t-1} + 0.95\epsilon_{t-2}$

Figure 5.8: Simulated ARMA processes with 0 mean and noise variance $\sigma^2 = 1$. The first one, for example, is obtained by the matlab commands `Z=randn(1,n)` and `X=filter([1 -0.4 +0.95],[1 0.4 -0.45],Z)`.

Which of these matlab scripts produce a sample $\boldsymbol{X}$ of an ARMA process ?

A. X=filter([1 ; -0.4],[1;0.4],randn(1,n))
B. X=filter([1 ; 0.4],[1;-0.4],randn(1,n))
C. A and B
D. None
E. I don’t know

ARMA Processes are Gaussian (non iid)

ARMA Process as a Gaussian Process Since an ARMA process is defined by linear transformation of a gaussian process ϵ_t it is a gaussian process. Thus it is entirely defined by its mean $\mathbb{E}(X_t) = \mu$ and its covariance. Its covariance can be computed in a number of ways, the simplest is perhaps obtained by noticing that

$$X_t = \mu + h_0\epsilon_t + \ldots + h_{t-1}\epsilon_1 \tag{5.24}$$

where h is the impulse response of the filter in Eq.(5.23). Note that, with our convention, $h_0 = 1$. It follows that for $t \geq 1$ and $s \geq 0$:

$$\text{cov}(X_t, X_{t+s}) = \sigma^2 \sum_{j=0}^{t-1} h_j h_{j+s} \tag{5.25}$$

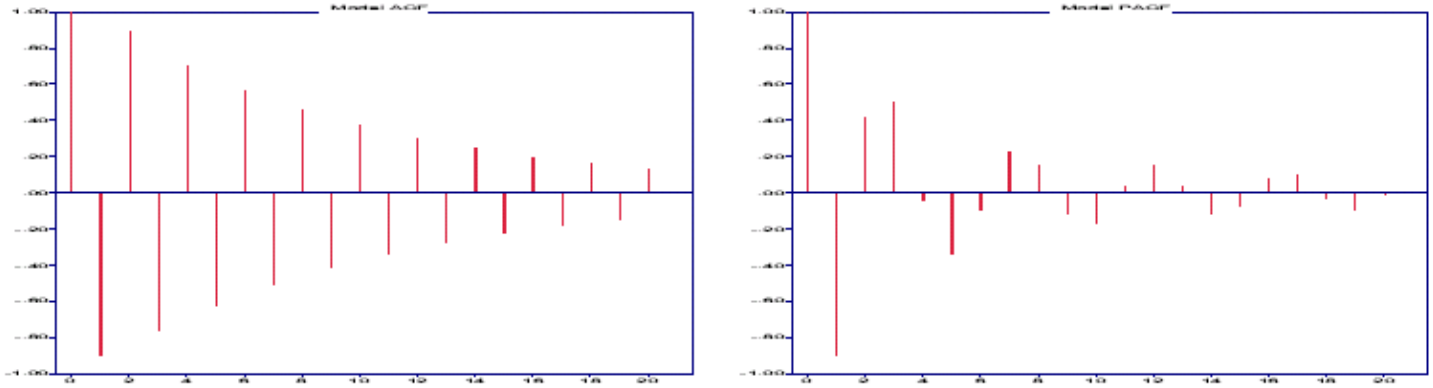
For large t

$$\text{cov}(X_t, X_{t+s}) \approx \gamma_s = \sigma^2 \sum_{j=0}^{\infty} h_j h_{j+s} \tag{5.26}$$

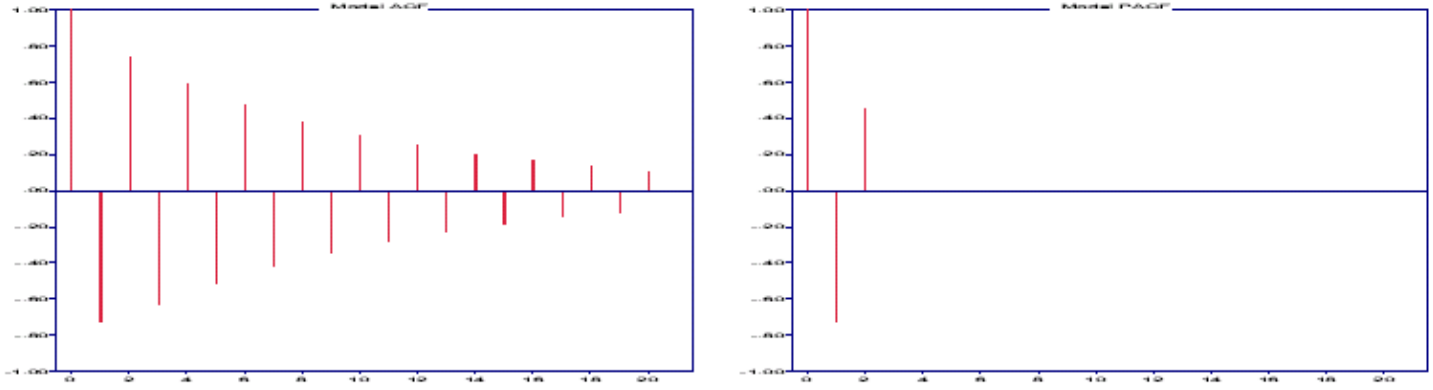
The convergence of the latter series follows from the assumption that the filter is stable. Thus, *for large t, the covariance does not depend on t. More formally, one can show that an ARMA* process with Hypothesis 5.1 is asymptotically stationary [19, 97], as required since we want to model stationary data[3].

$$\mathrm{var}(X_t) \approx \sigma^2 \sum_{j=0}^{\infty} h_j^2 = \sigma^2 (1 + \sum_{j=1}^{\infty} h_j^2) \geq \sigma^2$$

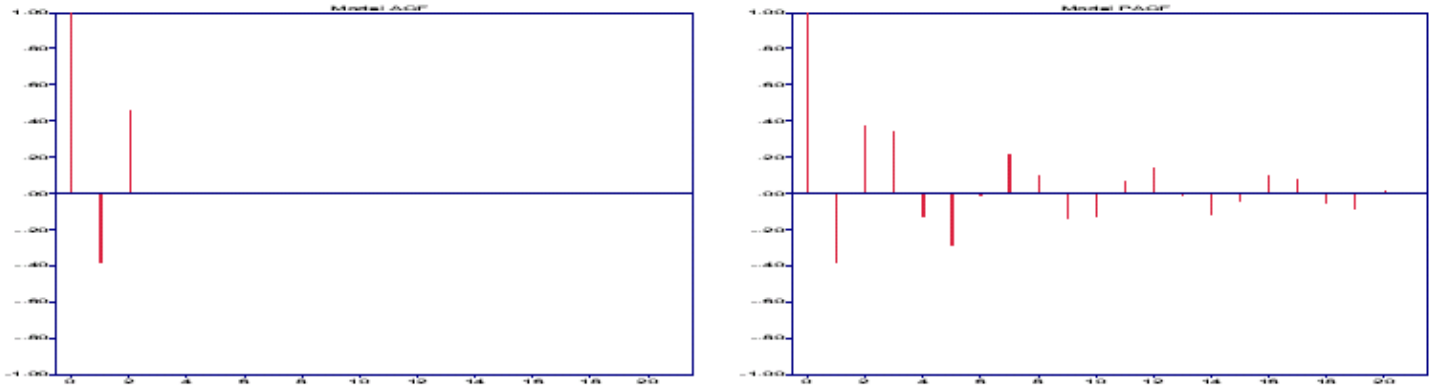
The *Auto-Correlation Function* (ACF) is defined as $\rho_t = \gamma_t / \gamma_0$. [5]. The ACF is quantifies departure from an iid model; indeed, for an iid sequence (i.e. $h_1 = h_2 = ... = 0$), $\rho_t = 0$ for $t \geq 1$. The ACF can be computed from Eq.(6.26) but in practice there are more efficient methods that exploit Eq.(6.23), see [36], and which are implemented in standard packages. One also sometimes uses the *Partial Auto-Correlation Function* (PACF), which is defined in Section A.5.2 as the residual correlation of X_{t+s} and X_t, given that $X_{t+1}, ..., X_{t+s-1}$ are known.[6]



(a) ARMA(2,2) $X_t = -0.4X_{t-1} + 0.45X_{t-2} + \epsilon_t - 0.4\epsilon_{t-1} + 0.95\epsilon_{t-2}$



(b) AR(2) $X_t = -0.4X_{t-1} + 0.45X_{t-2} + \epsilon_t$



(c) MA(2) $X_t = \epsilon_t - 0.4\epsilon_{t-1} + 0.95\epsilon_{t-2}$

ARIMA Process

Y_t is called an ARIMA process if $X = LY$ is an ARMA process, where L is a combination of differencing and deseasonalizing filters

How to fit an ARIMA process ?

- Apply differencing filters until appears stationary
- Fit the differenced process $X = LY$ using the ARMA fitting procedure (Thm 5.2, Matlab's **armax**);
- Check ACF of residuals; residuals are $\epsilon_t = X_t - \hat{X}_{t-1}(1)$ (innovation formula)
- Be careful with overfitting problem – use AIC or BIC; ACF of X may give an idea of order

Fitting an ARMA process is a non-linear optimization problem

Usually solved by iterative, heuristic algorithms,
may converge to a local maximum
may not converge

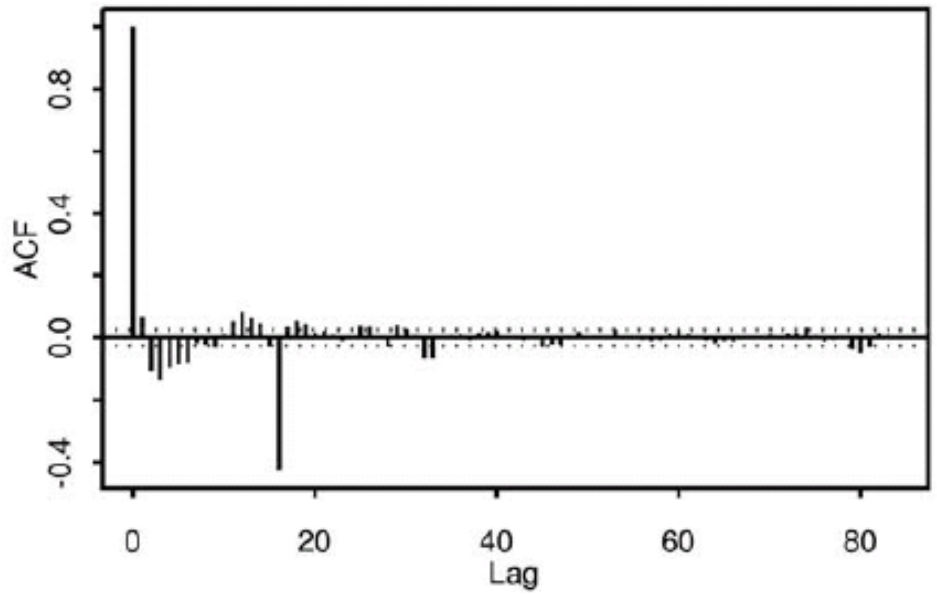
Some simple, non MLE, heuristics exist for AR or MA models

Ex: fit the AR model that has the same theoretical ACF as the sample ACF

Common practice is to bootstrap the optimization procedure by starting with a "best guess"

AR or MA fit, using heuristic above

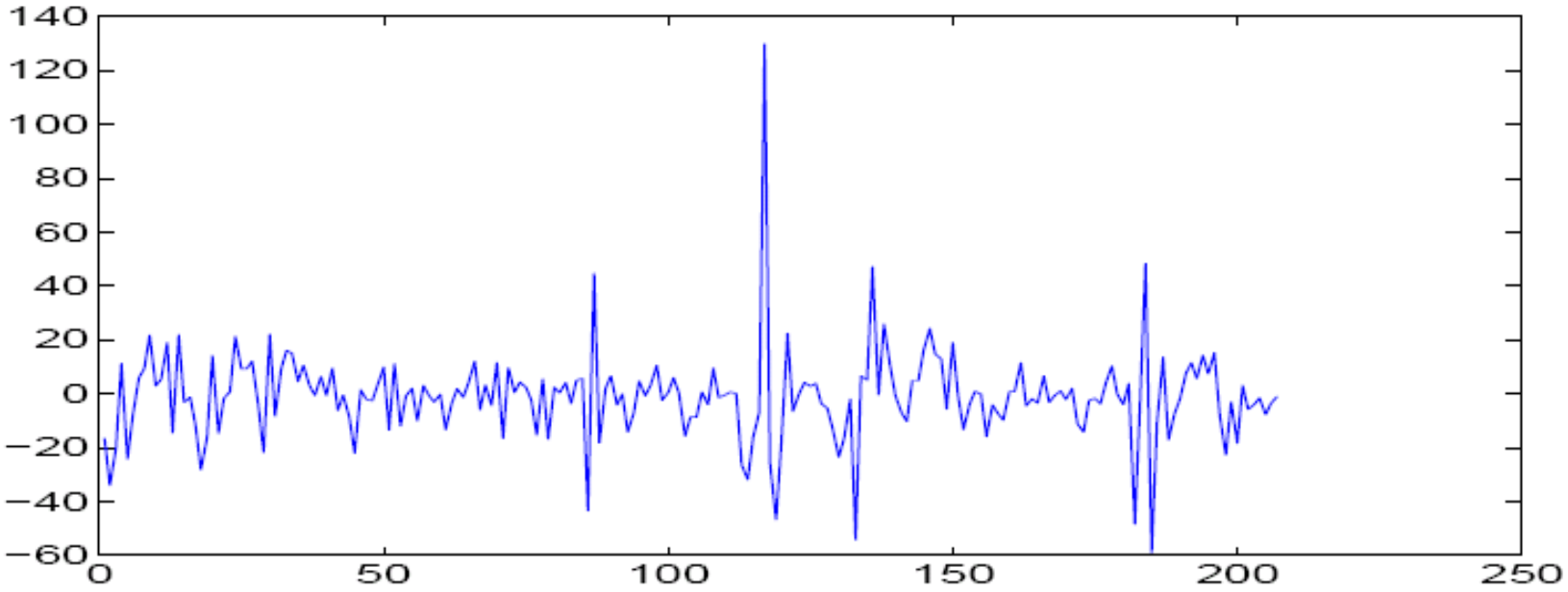
Example



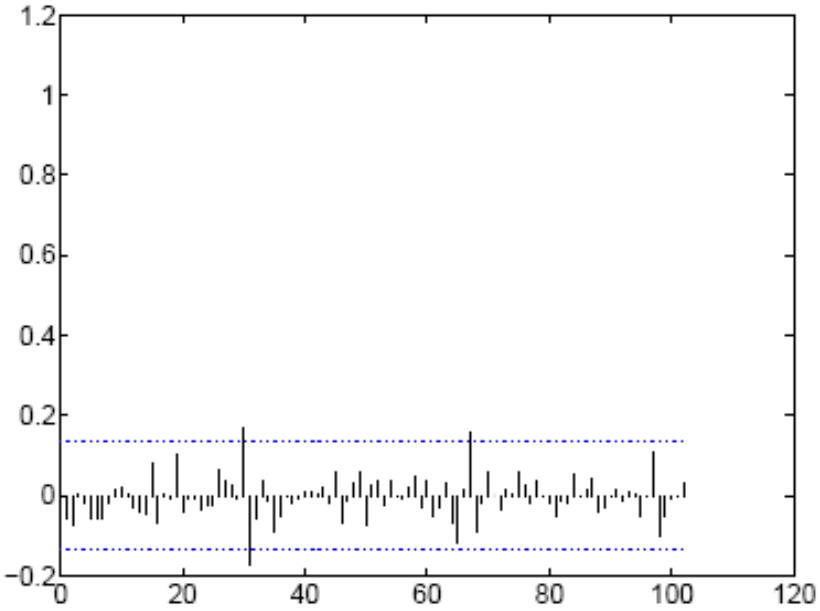
EXAMPLE 5.3: INTERNET TRAFFIC, CONTINUED. The differenced data in Figure 5.10 appears to be stationary and has decaying ACF. We model it as a 0 mean ARMA(p, q) process with $p, q \leq 20$ and fit the models to the data. The resulting models have very small coefficients A_m and C_m except for m close to 0 or above to 16. Therefore we re-fit the model by forcing the parameters such that

$$\begin{aligned} A &= (1, A_1, \ldots, A_p, 0, \ldots, 0, A_{16}, \ldots, A_{16+p}) \\ C &= (1, C_1, \ldots, C_p, 0, \ldots, 0, C_{16}, \ldots, C_{16+q}) \end{aligned}$$

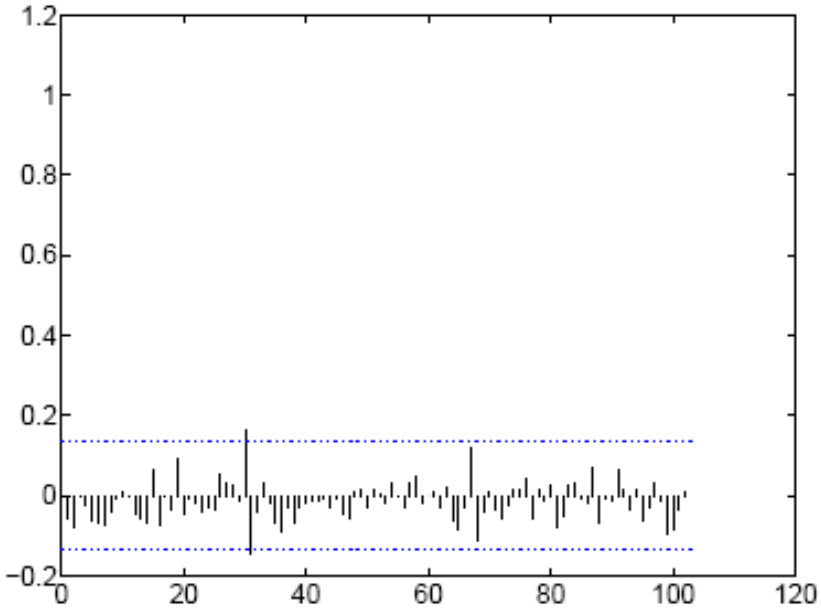
for some p and q. The model with smallest AIC in this class is for $p = 1$ and $q = 3$.



(b) Residuals



(c) ACF of Residuals



(d) PACF of Residuals

Forecasting with an ARIMA Process Y_t

By composition of filters, $Y = L^{-1}X = L^{-1}F\epsilon$ where F is the filter of the ARMA process and L is the differencing filter. Using the impulse response of $L^{-1}F$ and its inverse we obtain formulas similar to those we saw previously. See Prop 5.4 and forecast-exercise

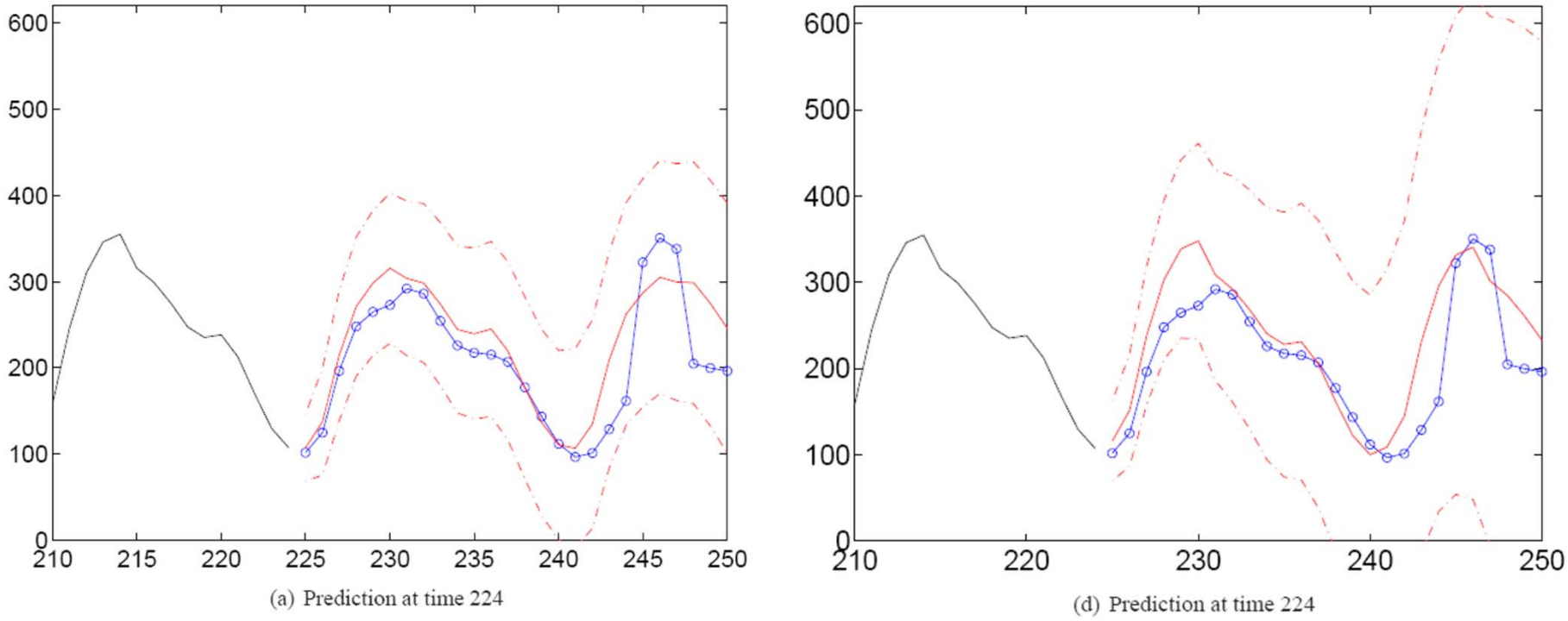
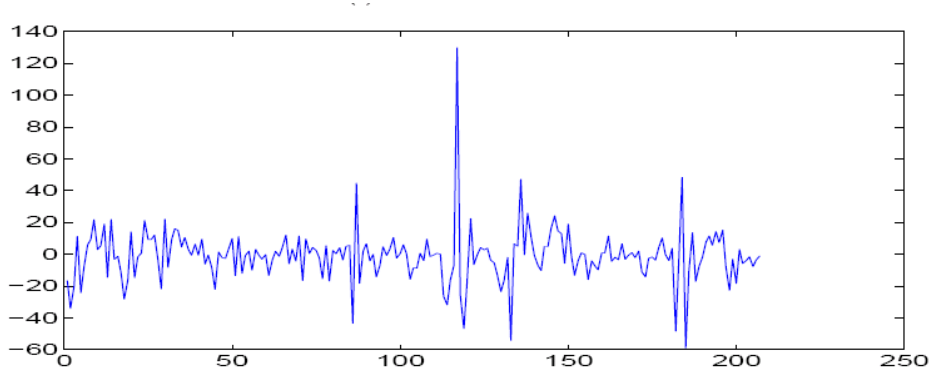
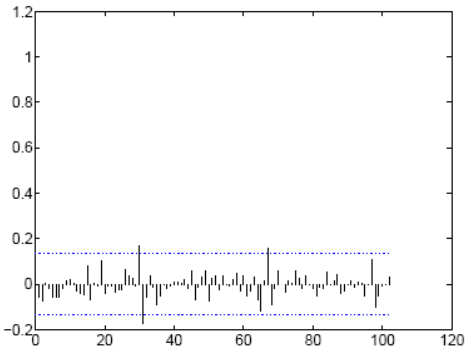


Figure 6.7: Differencing filters Δ_1 and Δ_{16} applied to Example 6.1 (first terms removed). The forecasts are made assuming the differenced data is iid gaussian with 0 mean. o = actual value of the future (not used for fitting the model).

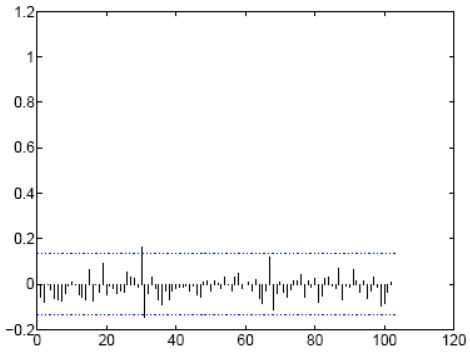
Improve Confidence Interval If Residuals are not Gaussian (but appear to be iid)



(b) Residuals



(c) ACF of Residuals



(d) PACF of Residuals

Assume residuals are not gaussian but are iid

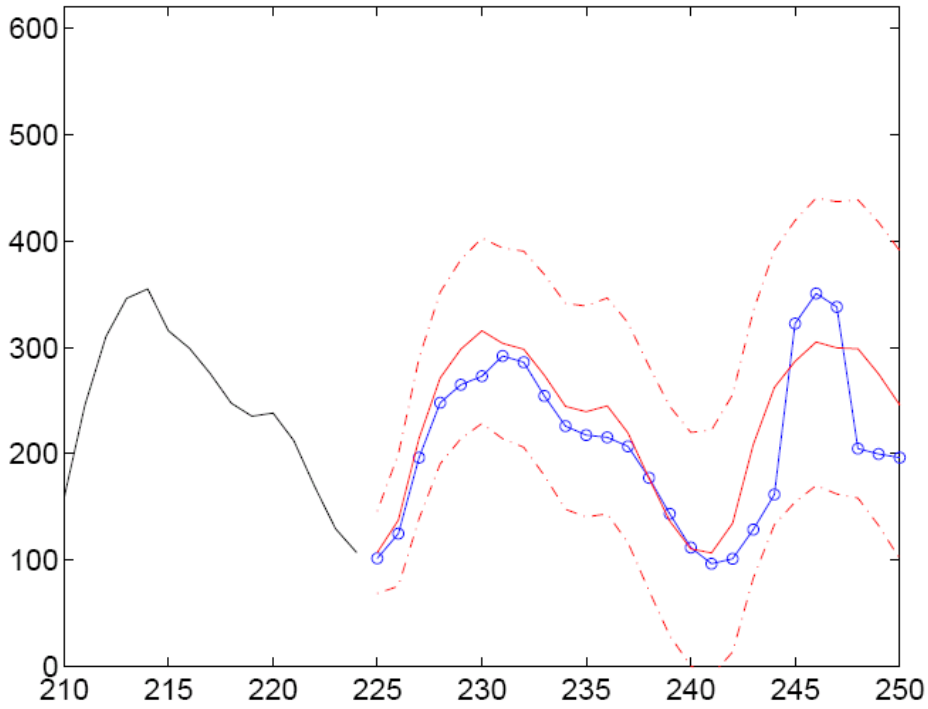
How can we get prediction intervals ?

Bootstrap by sampling from residuals

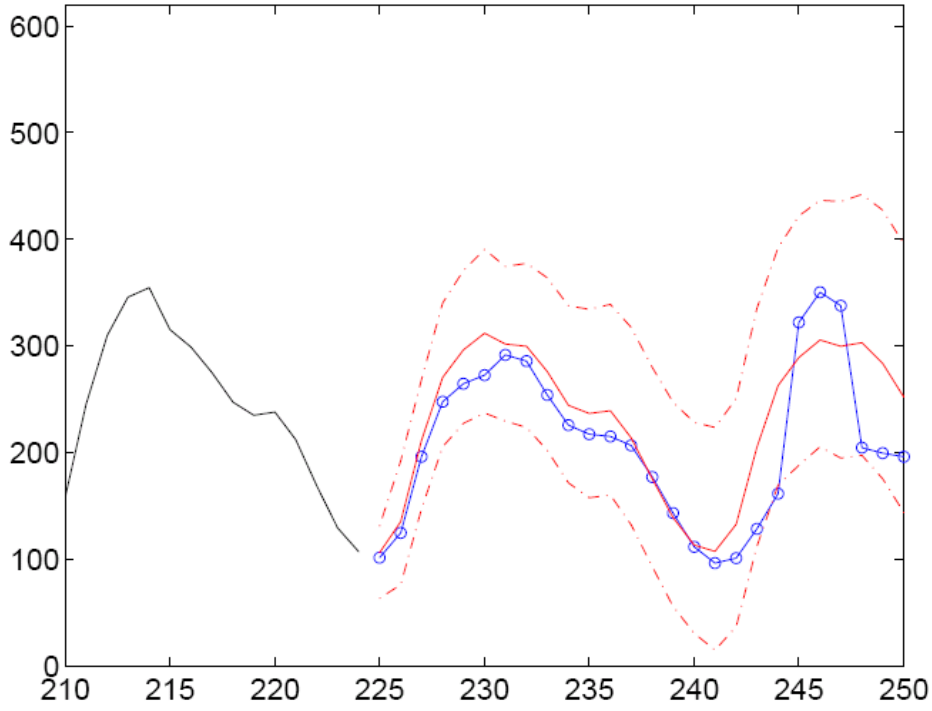
Algorithm 3 Monte-Carlo computation of prediction intervals at level $1-\alpha$ for time series Y_t using re-samplig from residuals. We are given: a data set Y_t, a differencing and de-seasonalizing filter L and an ARMA filter F such that the residual $\epsilon = F^{-1}LY$ appears to be iid; the current time t, the prediction lag ℓ and the confidence level α. r_0 is the algorithm's accuracy parameter.

1: $R = \lceil 2\, r_0/\alpha \rceil - 1$ ▷ For example $r_0 = 25$, $R = 999$
2: compute the differenced data $(x_1, \ldots, x_t) = L(y_1, \ldots, y_t)$
3: compute the residuals $(e_q, \ldots, e_t) = F^{-1}(x_q, \ldots, x_t)$ where q is an initial value chosen to remove initial inaccuracies due to differencing or de-seasonalizing (for example $q =$ length of impulse response of L)
4: **for** $r = 1 : R$ **do**
5: draw ℓ numbers with replacement from the sequence $(e_q, \ldots, e_t)$ and call them $\epsilon^r_{t+1}, ..., \epsilon^r_{t+\ell}$
6: let $e^r = (e_q, \ldots, e_t, \epsilon^r_{t+1}, ..., \epsilon^r_{t+\ell})$
7: compute $X^r_{t+1}, \ldots, X^r_{t+\ell}$ using $(x_q, \ldots, x_t, X^r_{t+1}, \ldots, X^r_{t+\ell}) = F(e^r)$
8: compute $Y^r_{t+1}, \ldots, Y^r_{t+\ell}$ using Proposition 5.1 (with X^r_{t+s} and Y^r_{t+s} in lieu of $\hat{X}_t(s)$ and $\hat{Y}_t(s)$)
9: **end for**
10: $\left(Y_{(1)}, ..., Y_{(R)}\right) = \text{sort}\left(Y^1_{t+\ell}, ..., Y^R_{t+\ell}\right)$
11: Prediction interval is $[Y_{(r_0)}\ ;\ Y_{(R+1-r_0)}]$

With gaussian assumption



With bootstrap from residuals



10. Other

We have seen a few forecasting recipes

- regression models
- use of differencing filters to make noise stationary
- use of ARMA models to make noise iid
- use of bootstrap

This can be combined or extended. For example: linear regression with ARMA noise

Linear Regression with ARMA Noise

Assume a linear regression model

$Y_t = \sum_i \beta_i \, x_t^i + \epsilon_t$ where we find that ϵ_t does not look iid at all.

We can model ϵ_t as an ARMA process and obtain

$Y_t = \sum_i \beta_i \, x_t^i + F w_t$ where F is an ARMA filter and w_t is iid $N(0, \sigma^2)$

Apply the inverse filter and obtain a linear regression model

$$(F^{-1}Y)_t = \sum_i \beta_i \left(F^{-1}x^i\right)_t + w_t, \qquad \text{with } w_t \sim \text{iid } N(0, \sigma^2)$$

If we know F we can estimate β; if we know β we can estimate F

$\Rightarrow$ iterate and hope it converges

Prediction formulae can be obtained using the calculus of filters exactly as we did above.

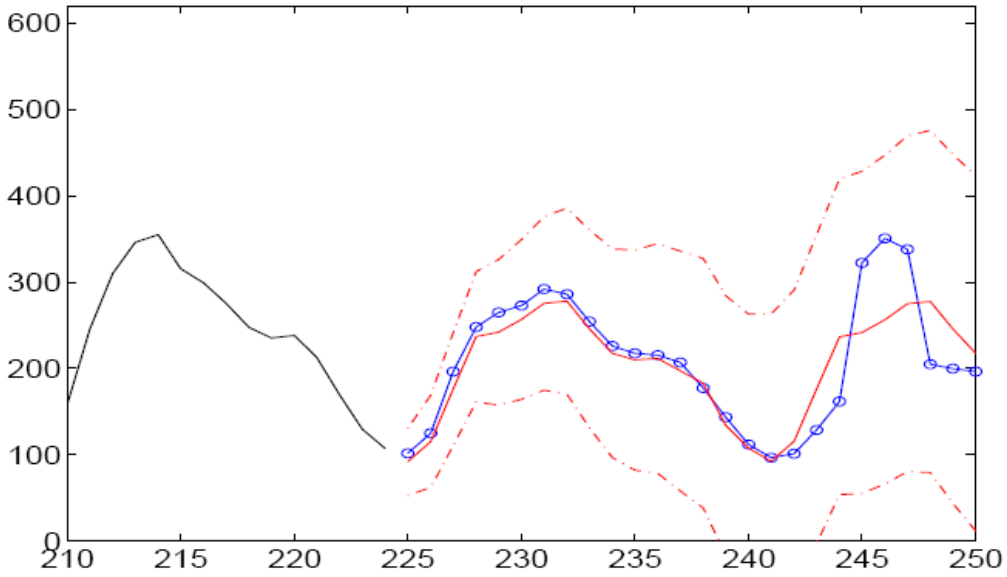
Sparse ARMA Models

Problem: avoid many parameters when the degree of the A and C polynomials are high

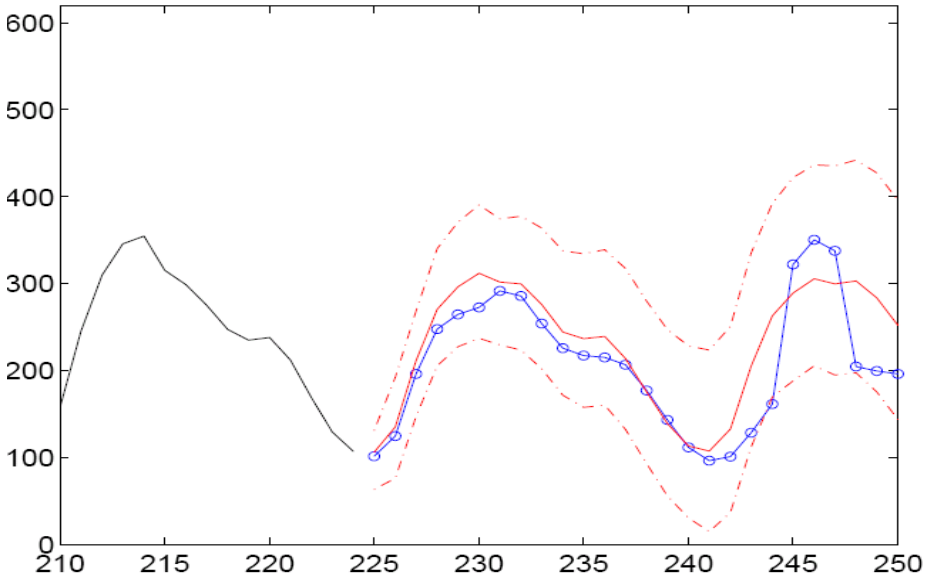
Based on heuristics

- Multiplicative ARIMA, constrained ARIMA
- Holt Winters

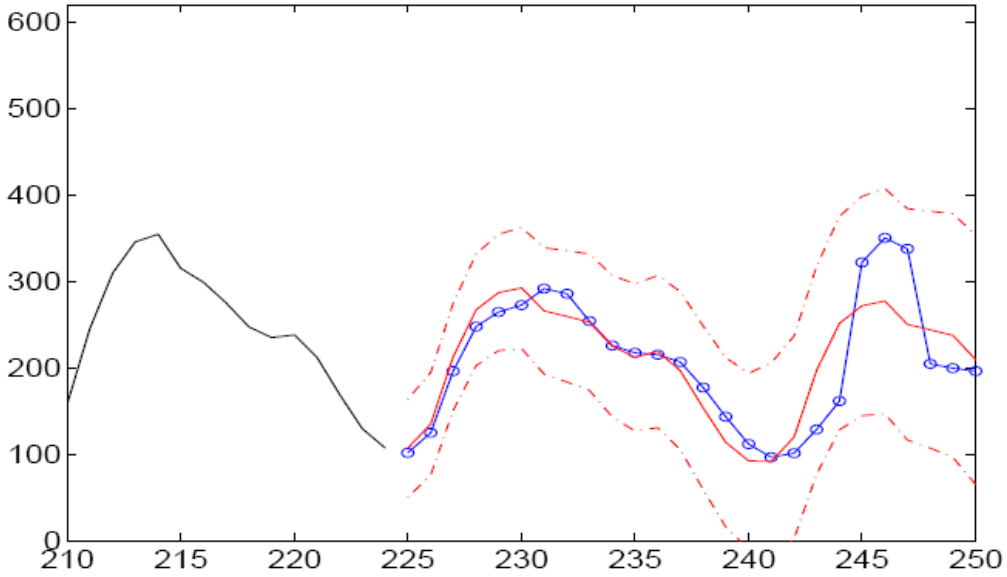
See section 5.6



(a) Roberts Model



Constrained ARIMA



(b) Holt Winters Additive Seasonal Model

Sparse models give less accurate predictions but have much fewer parameters and are simple to fit.