AR/MA/ARMA models

Susan Thomas

September 11, 2009

Goals

- Behaviour of ACF/PACF for AR/MA models
- Introducing ARMA models
- Understanding information in ARMA models, stationarity and invertibility conditions on their parameters.
- Homework!

Recap:

- Simplest process: white noise

$$x_t = \epsilon_t; \qquad \epsilon_t \sim D(\alpha_0, \sigma^2)$$

- First line of stochastic models: linear stationary models.
  - AutoRegressive process:

    $$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \ldots + \alpha_p y_{t-p} + \epsilon_t$$

    where $\epsilon_t \sim$w.n with $\mu_\epsilon = 0$.
  - Moving Average process:

    $$y_t = \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t$$

    where $\epsilon_t \sim$w.n. with $\mu_\epsilon = 0$.

Behaviour of AR vs. MA models of the same coefficient

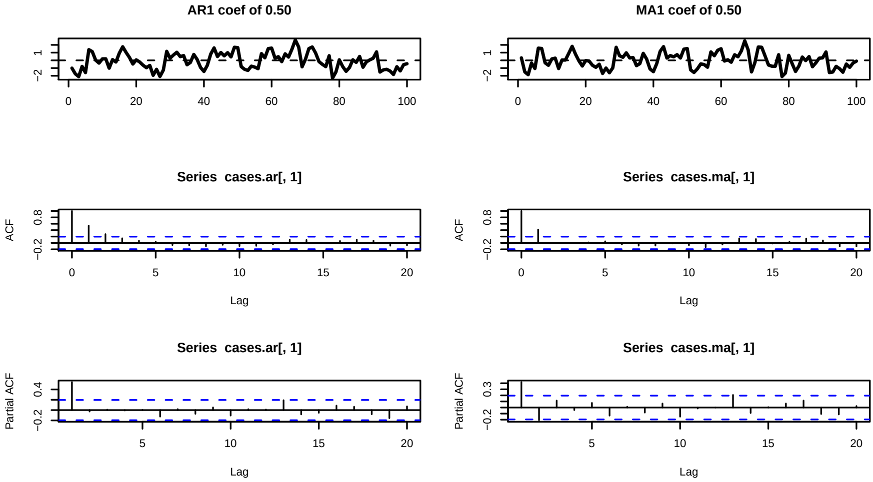
R-code generating data for AR(1)/MA(1), coeff = 0.5

```
# Setup our innovations.
startoff <- rnorm(200000)                    # Used by arima.sim for burn-in
ehacked <- e <- rnorm(100)
ehacked[50] = 5                             # Put one big shock into ehacked

# Comparing AR/MA models

make.ar.ma.series <- function(cff, e, startoff) {
  xar <- arima.sim(n=100, list(ar=cff), innov=e, start.innov=startoff)
  xma <- arima.sim(n=100, list(ma=cff), innov=e, start.innov=startoff)
  list(x=xar,y=xma)
}

cases.ar <- cases.ma <- NULL
coeffvalues <- c(0.5,0.75,0.98)
for (cff in coeffvalues) {
  cat("Doing ", cff, "\n")
  tmp <- make.ar.ma.series(cff, e, startoff)
  cases.ar <- cbind(cases.ar, tmp$x)
  cases.ma <- cbind(cases.ma, tmp$y)
}
```

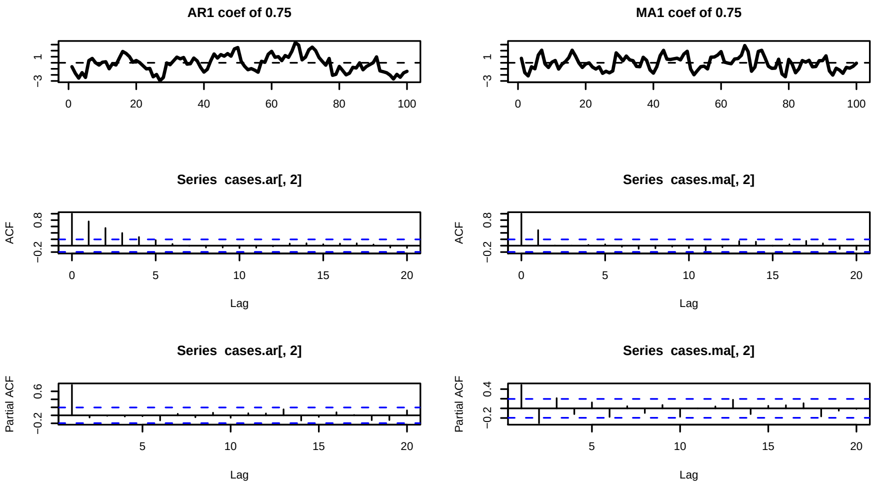
R-code generating data for AR(1)/MA(1), coeff = 0.5

```
pdf("pix/arvsma-0.5.pdf", pointsize=8, width=5.6, height=3.2)

hilo <- range(rbind(cases.ar[,1], cases.ma[,1]))
par(mfrow=c(3,2))
  plot(1:100, cases.ar[,1], type="l", ylim=hilo,
     xlab="", ylab="", lwd=2,
     main=sprintf("AR1 coef of %.2f", coeffvalues[1]))
  abline(h=0, lty=2)
  plot(1:100, cases.ma[,1], type="l", ylim=hilo,
       xlab="", ylab="", lwd=2,
       main=sprintf("MA1 coef of %.2f", coeffvalues[1]))
  abline(h=0, lty=2)
  acf(cases.ar[,1])
  acf(cases.ma[,1])
  pacf(cases.ar[,1])
  pacf(cases.ma[,1])
```

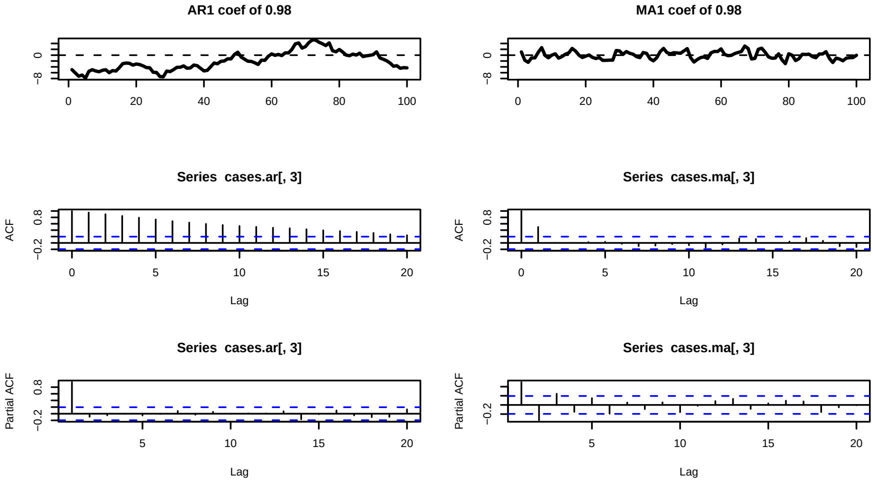
Example of AR/MA coeff = 0.5



Example of AR/MA coeff = 0.75



Example of AR/MA coeff = 0.98



Understanding the impact of information shocks on AR vs. MA processes

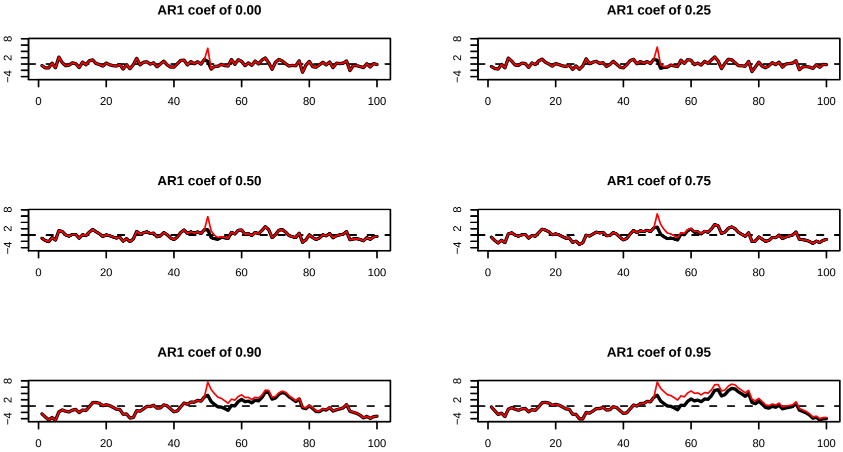
Example of information shocks on AR processes

```
make.a.series <- function(ar1, e, ehacked, startoff) {
  x <- arima.sim(n=100, list(ar=ar1), innov=e, start.innov=startoff)
  xhacked <- arima.sim(n=100, list(ar=ar1), innov=ehacked, start.innov
  list(x=x,xhacked=xhacked)
}

# Setup our innovations.
startoff <- rnorm(200000)                    # Used by arima.sim for burn-in
ehacked <- e <- rnorm(100)
ehacked[50] = 5                             # Put one big shock into ehacked

cases <- cases.hacked <- NULL
ar1values <- c(0, 0.25, 0.5, 0.75, 0.90, 0.95)
for (ar1 in ar1values) {
  cat("Doing ", ar1, "\n")
  tmp <- make.a.series(ar1, e, ehacked, startoff)
  cases <- cbind(cases, tmp$x)
  cases.hacked <- cbind(cases.hacked, tmp$xhacked)
}
```

Example of information shocks on AR processes

```
pdf("pix/ar-sensitivity.pdf", pointsize=8, width=5.6, height=3.2)

hilo <- range(rbind(cases, cases.hacked))
par(mfrow=c(3,2))
for (i in 1:6) {
  plot(1:100, cases[,i], type="l", ylim=hilo,
       xlab="", ylab="", lwd=2,
       main=sprintf("AR1 coef of %.2f", ar1values[i]))
  abline(h=0, lty=2)
  lines(1:100, cases.hacked[,i], col="red")
}
```

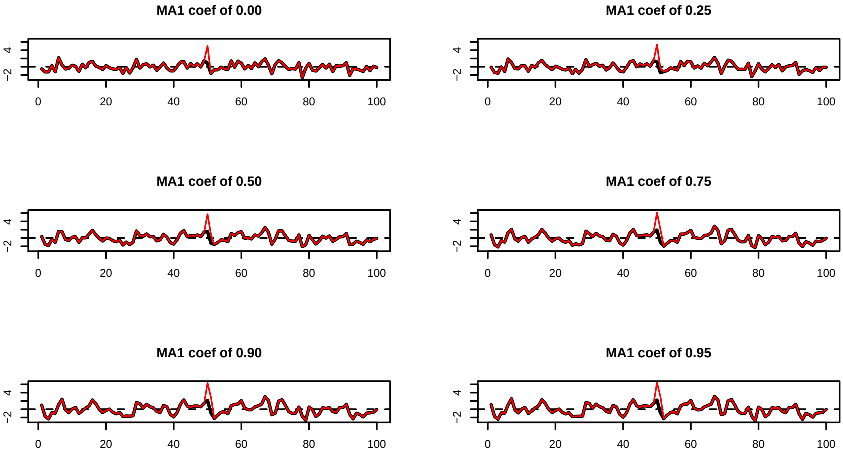
Example of information shocks on AR processes



Information in MA processes?

- What results would a similar simulation produce if the DGP was an MA process instead?

Example of information shocks on MA processes



Mixed AR/MA processes: ARMA models

Autoregressive moving average models – ARMA

- An ARMA DGP combines both an AR as well as an MA structure.
- Could be a parsimonious way to:
  - give different weights to innovations at different lags, and
  - retain a long lag structure of dependance.
- Generically, there could be p lags on the AR structure, q lags on the MA structure.
  ARMA(p, q):

$$
\begin{aligned}
y_t &= a + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} \\
&\quad + \theta_q \epsilon_{t-q} + \ldots + \theta_1 \epsilon_{t-1} + \epsilon_t \\
\epsilon_t &\sim \text{iid w.n.}
\end{aligned}
$$

Simplest ARMA model: ARMA(1,1)

- The simplest structure for an AR model is an ARMA(1,1)

$$\begin{aligned} y_t &= a + \phi y_{t-1} + \theta \epsilon_{t-1} + \epsilon_t \\ \epsilon_t &\sim \text{iid w.n.} \end{aligned}$$

- Alternative representation:

$$(1 - \phi L) y_t = a + (1 + \theta L) \epsilon_t$$

Effect of old information on y_t in ARMA(1,1)

$$
\begin{aligned}
y_t &= a + \phi L y_t + \epsilon_t + \theta\epsilon_{t-1} \\
&= a + \phi(a + \phi y_{t-2} + \epsilon_{t-1}) + \epsilon_t + \theta\epsilon_{t-1} \\
&= (1+\phi)a + \phi^2 y_{t-2} + (\theta+\phi)\epsilon_{t-1} + \epsilon_t \\
&= (1+\phi)a + \phi^2(a + \phi y_{t-3} + \epsilon_{t-2}) + (\theta+\phi)\epsilon_{t-1} + \epsilon_t \\
&= (1+\phi+\phi^2)a + \phi^3 y_{t-3} + \phi^2\epsilon_{t-2} + (\theta+\phi)\epsilon_{t-1} + \epsilon_t \\
&= \ldots \\
&= (1+\phi+\phi^2+\ldots+\phi^{T-1})a + \\
&\quad\ \phi^T y_0 + \phi^{T-1}\epsilon_1 + \ldots + \phi^2\epsilon_{t-2} + (\theta+\phi)\epsilon_{t-1} + \epsilon_t
\end{aligned}
$$

Stationarity properties of ARMA(1,1)

- Model: $y_t - \phi y_{t-1} = \epsilon_t - \theta \epsilon_{t-1}$
- The process is **stationary** if $0 \leq |\phi| < 1$.
  The process is **invertible** if $0 \leq |\theta| < 1$.

Autocovariances/Autocorrelations of ARMA(1,1)

- Multiply the model through by y_{t-k} and take expectations gives the following autocovariance structure:

$$\begin{aligned}
\gamma_k &= \phi_1 \gamma_{k-1} \quad \forall (k \geq 2) \\
\gamma_1 &= \phi_1 \gamma_1 - \theta_1 \sigma_\epsilon^2 \\
\gamma_0 &= \phi_1 \gamma_1 + \sigma_\epsilon^2 - \theta_1 \gamma_{y,\epsilon}^2(-1)
\end{aligned}$$

- This gives autocovariances:

$$\begin{aligned}
\gamma_0 &= \frac{(1 + \theta_1^2 - 2\phi_1\theta_1)}{1 - \phi_1^2} \sigma_\epsilon^2 \\
\gamma_1 &= \frac{(1 - \phi_1\gamma_1)(\phi_1 - \theta_1)}{1 - \phi_1^2} \sigma_\epsilon^2 \\
\gamma_k &= \phi_1 \gamma_{k-1} \quad \forall (k \geq 2)
\end{aligned}$$

- Divide through by γ_0 to get autocorrelations, AC(k).

The impact of old information in ARMA models analysed

- From the above, we see that the ACF of the ARMA(1,1) model will have a decaying form. (Given that the stationarity condition implies $|\phi_1| < 1$)
- The decay will be smooth if ϕ_1 is positive, the decay will be smooth.
  If ϕ_1 is negative, the ACF will alternate.
- The sign of ρ_1 will depend upon the sign of $(\theta_1 - \phi_1)$.

Behaviour of ACFs of a generic ARMA(p, q) process

- Autocovariance function for ARMA(p,q):

  $$\gamma_k = \phi_1 \gamma_{k-1} + \ldots + \phi_p \gamma_{k-p} + \gamma_{y,\epsilon}(k) - \theta_1 \gamma_{y,\epsilon}(k-1) - \ldots - \theta_q \gamma_{y,\epsilon}(k-q)$$

  Solving for this gives these general observations:
  - The order of the MA process, q, determines how many ACs depend upon both the ϕ and the θ values.
    These are $\rho_1, \ldots, \rho_q$.
  - The order of the AR process, p, is the number of ACs beyond which the pattern of the ACs are a fixed pattern.
    These are $\rho_q, \ldots, \rho_{q-p+1}$.
  - Thus, if the MA order is less than the AR order, the ACF will be a decaying process.
  - If the MA order is much larger than the AR order, then the first $(q - p + 1)$ ACs will not follow the repeating pattern.
  - If $\theta >> \phi$, and/or ϕ is not large, then the process will tend to have low persistence and appear to be more like an MA process.
  - If $\phi >> \theta$, and/or ϕ is closer to one, then the process will tend to have high persistence.

HW, due by Friday next

HW1: Work out the form of the autocovariance/partial autocovariance functions

- Chapter 3 in edition 5 of **Time Series Analysis: Forecasting and Control**, by *Box et al.* is on linear stationary stochastic processes.
- There is a subsection (3.2.2 in the 5^{th} edition) on the Autocovariance/Partial Autocovariance functions. Check the index for “Yule-Walker” equations.
- Work through this for all the AR models with lags = 1,2,3.

HW2: Write a program to simulate an ARMA(1,1) data series

- For the following sets of AR-MA coefficients:
  1. ar(1), ma(1): $0.2, 0.8$
  2. ar(1), ma(1): $0.5, 0.5$
  3. ar(1), ma(1): $0.8, 0.2$
- Calculate the expected values of the ACs at $k = 1, 2, 3$.
- Plot the ACFs of the above three processes and verify that the simulated data generates theoretically expected values of the ACs.

HW3: Data analysis of sample.rda

- There are two datasets in the file "sample.rda".
  "Daily" contains daily data.
  "Monthly" contains monthly data.
  These are both "zoo" objects.
- For each data series,
  - create the log series
  - create the first differences of the log series.
- For each series
  1. Plot the series, the acf, the pacf (all three graphs on one page for a given series)
  2. From the patterns, deduce whether the series is:
     1. AR/MA/ARMA
     2. The order of the series